



UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH  
Escola d'Enginyeria de Barcelona Est

TREBALL FI DE MÀSTER

**Màster en Enginyeria Química**

**DESENVOLUPAMENT DE SIMULACIÓ, OPTIMITZACIÓ I  
COMUNICACIONS PER A PROCESSOS QUÍMICS**



**Memòria i Annexos**

<b>Autor:</b>	Joel Nevado Formenti
<b>Director:</b>	Moisés Graells Sobré
<b>Convocatòria:</b>	Juny 2017



## Resum

En aquest projecte s'ha realitzat el desenvolupament d'un model i la simulació d'un cas d'optimització d'una cadena de subministrament, així com l'ontologia que permet englobar el problema, utilitzant diferents eines de comunicació, com OPC, per a poder establir el contacte entre els diferents elements i que siguin capaços de compartir el coneixement i la informació.

El principal motiu d'aquest projecte és el desenvolupament d'una base de coneixement genèrica a partir d'un problema concret, que permeti l'aplicabilitat d'aquesta base en qualsevol problema del mateix caire.

## Resumen

En este proyecto se ha realizado el desarrollo de un modelo y simulación de un caso de optimización de cadena de suministro, así como la ontología que permite englobar el programa, utilizando diferentes herramientas de comunicación, como OPC, para poder establecer el contacto entre los distintos elementos y que sean capaces de compartir conocimiento e información.

El principal motivo de este proyecto es el desarrollo de una base de conocimiento genérica a partir de un problema concreto, que permita la aplicabilidad de esta base en cualquier problema del mismo tipo.

## **Abstract**

In this project, the development of a model and simulation for an optimization supply chain problem has been developed, as well as an ontology that generalizes the problem, using communication tools, such as OPC, to establish a contact between the different parts, so they can share knowledge and information.

The main reason of this project is the development of a base for generic knowledge based on a concrete problem, that allows the applicability of this base to any similar problem.





## Agraïments

Agrair en primer lloc al Dr. Moisès Graells i al Dr. Antonio Espuña, per la seva ajuda, així com per la seva vocació en el projecte. Agrair, també, a CEPIMA, per la seva col·laboració mitjançant les seves eines, així com per les hores d'ajuda.

Als companys del CIAO, per les hores que hem invertit allà treballant.

I als amics i a la família, els quals són els que més m'han hagut d'aguantar durant tot aquest temps.







## Glossari

COM: *Component Object Model* és una plataforma de Microsoft per components de Software que permet la comunicació entre processos i la creació dinàmica d'objectes.

DCOM: *Distributed Component Object Model* és una evolució del COM que permet treballar en diferents ordinadors gràcies a la comunicació entre aquests.

DDE: *Dynamic Data Exchange* és una tecnologia de comunicació entre aplicacions de Microsoft.

DLL: *Dynamic Link Library* és el terme que fa referència als arxius que són executats en un programa per ordre del sistema operatiu.

DCS: *Distributed Control System* és un sistema de control aplicat a processos complexos.

OLE: *Object Linking and Embedding* és la tècnica coneguda com incrustació i enllaçament d'objectes, que és la idea que hi ha darrere d'arrossegar i deixar anar un objecte amb el ratolí.

OPC: *OLE for Process Control* o *Open Platform Communications* és un estàndard de comunicació en el camp de control i supervisió de processos industrials basat inicialment en tecnologia Microsoft.

PAC: *Programmable Automatic Controller* és una tecnologia orientada al control automatitzat, disseny de prototipus i a la medicació.

PLC: *Programmable Logic Controller* és un controlador dissenyat per a programar i controlar processos seqüencials a temps real.

RTU: *Remote Terminal Unit* és el conjunt d'elements externs que complementen un SCADA, generalment utilitzat per obtenir informació.

SCADA: *Supervisory Control And Data Acquisition* és un sistema de control industrial per a la monitorització i control de processos.

UA: *Unified Architecture* és una variant del OPC que millora la comunicació amb altres sistemes operatius, millorant així també la seva funcionalitat.



# Índex

<b>RESUM</b>	<b>I</b>
<b>RESUMEN</b>	<b>II</b>
<b>ABSTRACT</b>	<b>III</b>
<b>AGRAÏMENTS</b>	<b>V</b>
<b>GLOSSARI</b>	<b>VII</b>
<b>1. PREFACI</b>	<b>1</b>
1.1. Motivació.....	1
1.2. Contextualització.....	1
1.2.1. Control i gestió de processos .....	1
1.2.2. Comunicació industrial .....	3
1.2.3. Ontologia .....	7
1.2.4. Estat de l'art.....	8
<b>2. INTRODUCCIÓ</b>	<b>9</b>
2.1. Objectius del treball .....	9
2.2. Abast del treball .....	9
<b>3. PRESA DE DECISIONS</b>	<b>13</b>
3.1. Suport ontològic.....	13
3.2. Simulador emprat .....	13
<b>4. DESENVOLUPAMENT</b>	<b>15</b>
4.1. Optimització .....	15
4.2. Ontologia .....	16
4.3. Simulació .....	22
<b>5. RESULTATS</b>	<b>25</b>
5.1. Optimització .....	25
5.2. Ontologia .....	26
5.3. Documentació .....	27
5.3.1. Implementació.....	27
5.3.2. Arquitectura.....	27
<b>CONCLUSIONS</b>	<b>29</b>



<b>TREBALL FUTUR</b>	<b>31</b>
<b>PRESSUPOST I/O ANÀLISI ECONÒMICA</b>	<b>33</b>
<b>BIBLIOGRAFIA</b>	<b>35</b>
<b>ANNEX 1: GUIA D'USUARI</b>	<b>37</b>
A1. Arquitectura .....	37
A2. Seqüència d'inici .....	37
A3. Seqüència d'Operació .....	39
<b>ANNEX 2: CODI MATLAB</b>	<b>41</b>

# 1. Prefaci

## 1.1. Motivació

Actualment, l'ús d'eines que ajudin als operadors a resoldre grans quantitats de càlculs és molt important, i, moltes d'aquestes vegades, la informació és reaprofitable per altres eines o operadors. És per això que s'ha considerat la necessitat de desenvolupar una eina capaç de reunir i organitzar tota la informació del procés, ja sigui referent a instrumentació, simulacions o altres elements mitjançant la comunicació OPC i exposar-la per a que sigui de lliure disposició per a tots els que la vulguin utilitzar.

Paral·lelament, sorgeix la necessitat que sigui l'ordinador qui tingui cada vegada més capacitats per resoldre els problemes. És a dir, la finalitat és que un ordinador no sigui només capaç de resoldre un problema ja plantejat, sinó a més, ser capaç d'entendre'l i plantejar-lo. Actualment, els ordinadors són només capaços de resoldre problemes, però si es vol arribar al punt en que un ordinador sigui capaç de plantejar i resoldre un problema, cal primer que l'ordinador sigui capaç d'aprendre i analitzar aquella informació ja existent.

És per això que s'ha considerat que l'ús d'una ontologia és adient en aquest cas, ja que permet reunir una quantitat suficientment gran d'informació i, a més, estructurar-la, d'acord als criteris de l'usuari, per a que sigui comprensible, sent, a més, una eina que també pot comunicar-se amb altres aplicacions, facilitant la gestió i control del procés i, finalment, genera una connexió entre conceptes i permet desenvolupar una base de coneixement, similar a un mapa conceptual, que és el pas necessari per a permetre a un ordinador simular el pensament humà, o el que és el mateix, desenvolupar una intel·ligència artificial.

## 1.2. Contextualització

### 1.2.1. Control i gestió de processos

S'entén per control i gestió de processos el conjunt d'eines que permeten tractar la informació referent als respectius processos. El control és el conjunt d'operacions orientades a supervisar l'estat d'un sistema, mentre que la gestió és el conjunt d'operacions orientades a seleccionar, interpretar, analitzar i distribuir la informació

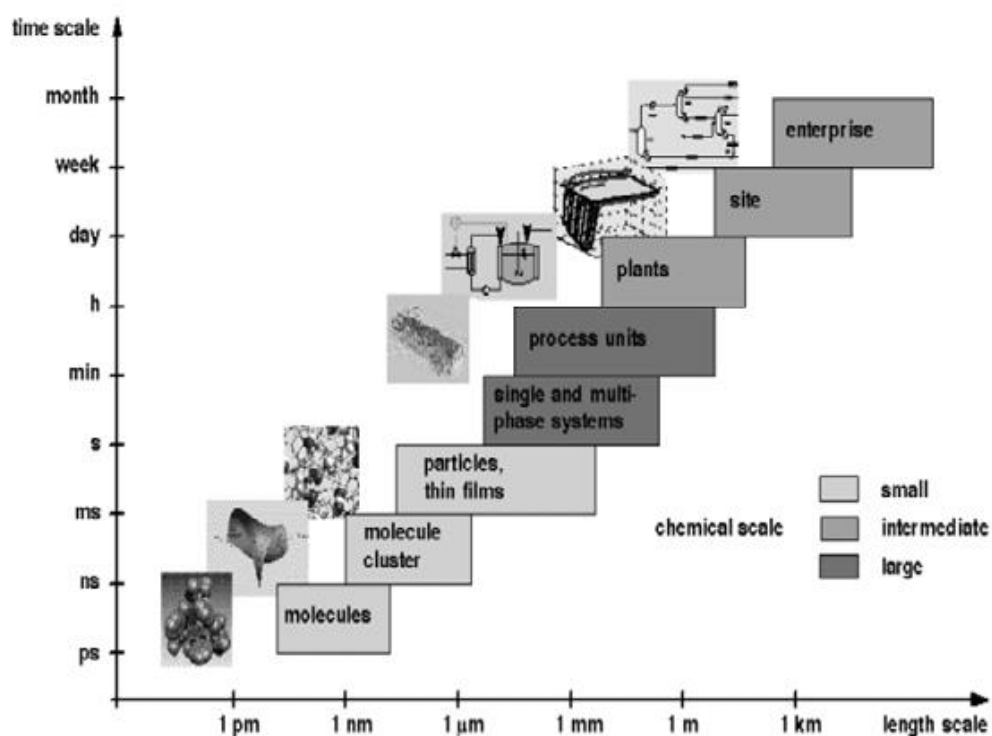
Hi ha un grup de recerca (Institution of Chemical Engineers n.d.) que diu que els sistemes de control i procés de la informació per l'operació de processos de plantes inclouen des de instrumentació, gestió de la informació i control per ordinador fins a modelització, simulació i suport a la presa de decisions.

Venkat Venkatasubramanian (Venkatasubramanian 2009) engloba tots aquests conceptes en un sol terme: cyberinfraestructure, que la defineixen com la interconnexió de software, hardware, middleware, bases de dades i recursos humans mitjançant una xarxa.

D'altra banda, (Venkatasubramanian 2009) diu que per abordar els reptes de modelització i la informàtica, cal realitzar un canvi en el nucli de la infraestructura, canviant una infraestructura centrada en l'aplicació per una centrada en la informació

Relacionat amb el que s'ha citat fins ara, (Committee on Challenges for the Chemical Sciences in the 21st Century, National Research Council 2003) explica que tant l'enginyeria de processos com la química computacional produeixen nou enteniment i desenvolupament en nous procediments computacionals en àmbits de simulació, disseny i operació, des del nivell atòmic fins al nivell industrial.

Un exemple del significat d'això és la cadena de subministrament (figura 1), on partint d'un disseny d'un producte a nivell molecular, s'obté la mescla macroscòpica que, a la simulació del procés, intervé a les diferents operacions unitàries per a que, finalment, el resultat obtingut sigui distribuït al mercat.



**Figura 1.** Esquema de la cadena de subministrament. (Committee on Challenges for the Chemical Sciences in the 21st Century, National Research Council 2003)

Altrament, també hi ha publicacions que (Committee on Challenges for the Chemical Sciences in the 21st Century, National Research Council 2003) indiquen que un dels principals reptes de l'enginyeria de processos és incloure el disseny del producte com a part de l'enginyeria de processos, així com la integració de les diferents parts de la cadena de subministrament poden evolucionar en modelitzacions a diferents escales.

En resum, des de fa temps, ja s'està parlant sobre la importància de la informació en els processos químics i la importància en saber-la tractar i gestionar, fins al punt de poder considerar-la l'element central d'una infraestructura.

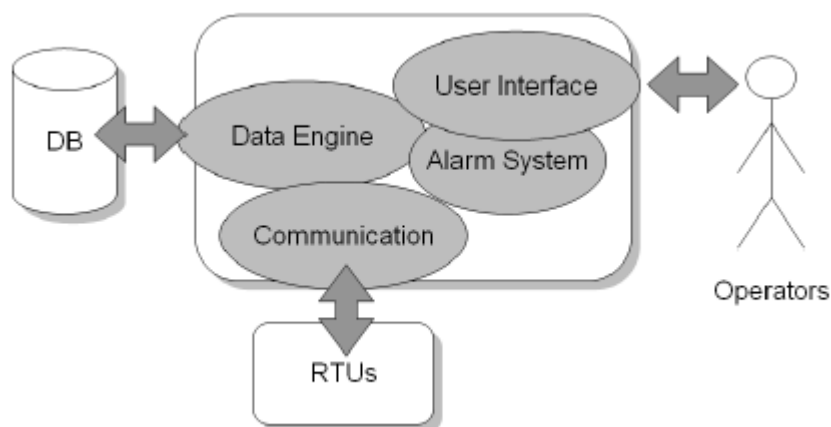
### 1.2.2. Comunicació industrial

La comunicació industrial és el conjunt d'eines i tècniques per transmetre informació d'un punt a un altre connectats entre sí. Aquesta connexió pot ser tant física com via internet. Les eines més utilitzades a la comunicació industrial són SCADA i OPC.

### Supervisory Control and Data Acquisition

El Supervisory Control And Data Acquisition (de l'anglès, SCADA) és un sistema de control industrial per a la monitorització i control de processos. Un SCADA sol estar compost per diferents elements, com un sistema d'alarmes, una base de dades i una interfície d'usuari. A més a més, es pot comunicar externament amb altres elements del sistema, anomenats Remote Terminal Unit (RTU), com poden ser un Programmable Logic Controller (PLC), un Distributed Control System (DCS) o un Programmable Automation Controller (PAC).

Un dels principals inconvenients d'un SCADA és que és únic per cada indústria i, per tant, comunicar-ne de diferents pot ser difícil.

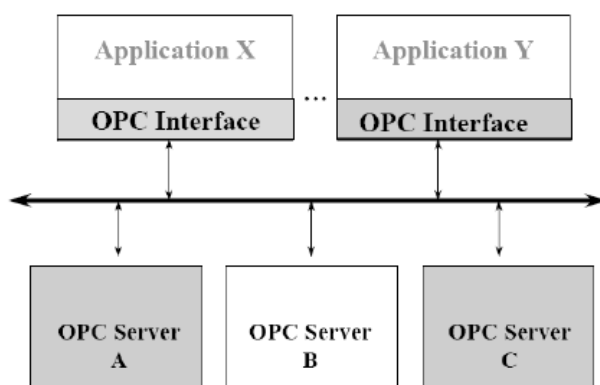


**Figura 2.** Esquema de SCADA i RTU. (Skeie 2013)

### **Object Link and Embedding for Process Control**

El Object Linking and Embedding for Process Control (OLE for Process Control o OPC) és un estàndard de comunicació en el camp de control i supervisió de processos industrials basat en tecnologia Microsoft. La principal raó de l'existència del OPC és el saber valors de dades i l'estat o qualitat d'aquestes, així com poder connectar diferents dispositius de control i gestió de dades.

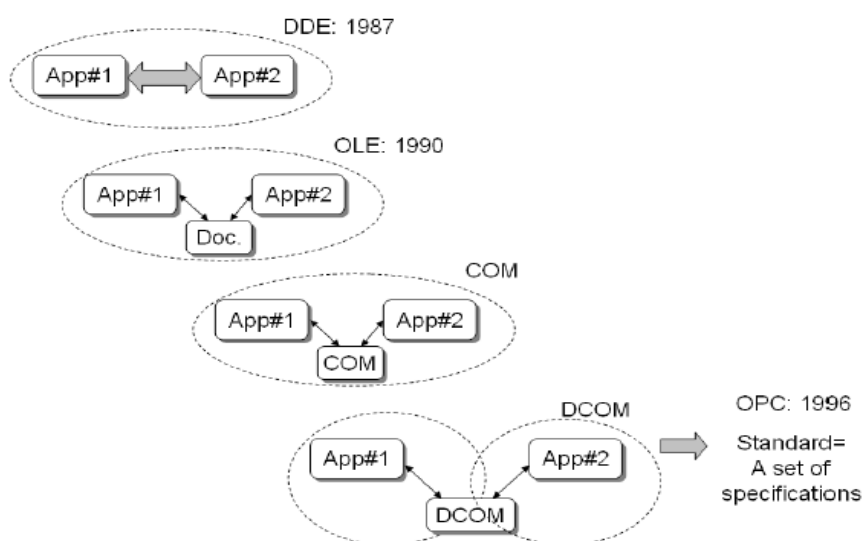




**Figura 3.** Esquema de OPC. (Skeie 2013)

El OPC surgeix a partir del Dynamic Data Exchange (DDE), una especificació creada el 1987 per establir una primera comunicació entre especificacions basades en Microsoft. Un dels principals inconvenients que tenia era l'amplada de banda, per la qual cosa va crear-se el protocol Object Linking and Embedding (OLE) el 1990, el qual podia mantenir connexió entre dos documents. A partir d'aquí, OLE va evolucionar fins a esdevenir una arquitectura per components de software, conegut també com a Component Object Model (COM), on els documents són tractats com objectes. Malgrat poder establir una comunicació entre diferents documents, aquesta estava limitada a un sol ordinador. Com a conseqüència, i per tal de solucionar aquest inconvenient, es va desenvolupar el Distributed Component Object Model (DCOM), que permetia la comunicació entre xarxes d'ordinadors. A partir d'aquí, l'ús del DCOM i el OLE va esdevenir en el que es coneix des de 1996 com OPC.

Recentment, la OPC foundation ha canviat el concepte de OPC de Object Linking and Embedding for Process Control a OPC classic, degut a l'aparició de noves versions de OPC que engloben les característiques que té el OPC Classic, provocant que el OPC classic, que va néixer fa tants anys, sigui obsolet.



**Figura 4.** Evolució del OPC. (Skeie 2013)

OPC es divideix en dos models de comunicació: el model Client/Servidor i el model Publicador/Subscriber.

**Taula 1.** Comparativa entre Client/Servidor i Publicador/Subscriber.

Client/Servidor	Publicador/Subscriber
El servidor és el propietari de la informació	Publicació cíclica per part del propietari de la informació
El client sol·licita al servidor la informació	Els clients s'han de subscriure al propietari
Diferents clients poden accedir al mateix servidor	Un cop realitzada la subscripció, els clients reben automàticament la informació
La seqüència d'informació sempre s'inicia quan el client demana al servidor la informació	

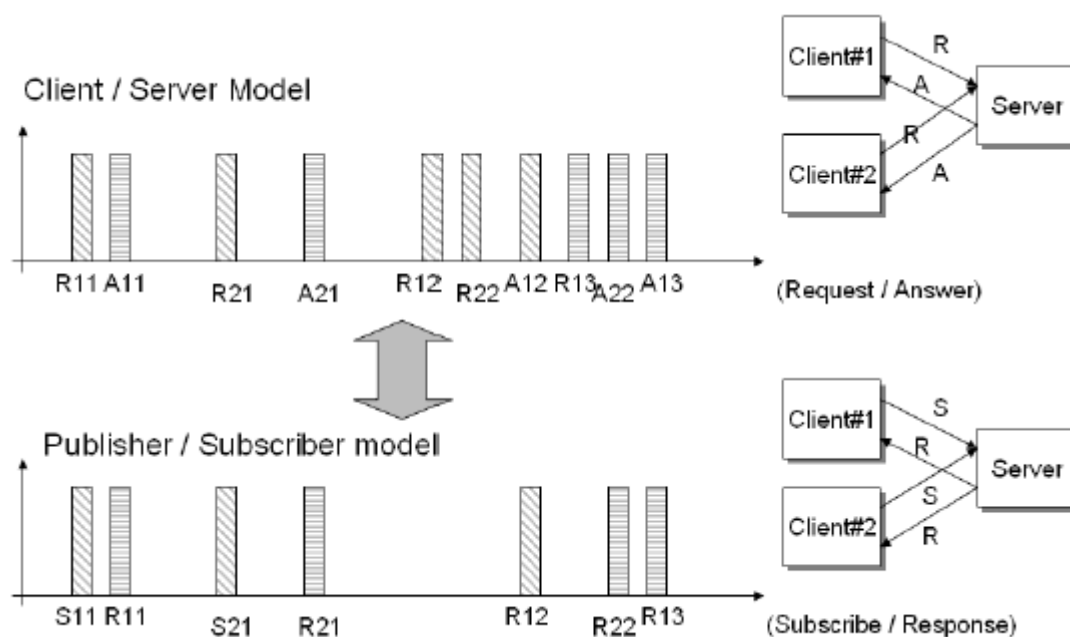


Figura 5. Comparativa entre els models. (Skeie 2013)

Per tal de solucionar els diferents que pot tenir el OPC (seguretat i número de servidors, entre altres), una nova estructura OPC, coneguda com *Unified Architecture* (UA), s'ha desenvolupat amb noves característiques:

- ús d'arxius XML i .NET de Microsoft en lloc de les *Dynamic Link Library* (DLL) que utilitza un dels components de OPC, DCOM, degut als problemes amb els *firewalls*.
- millor integració de les diferents especificacions del OPC. En lloc de tenir un servidor per cada especificació del OPC, s'intenta combinar per tal de reduir el nombre de servidors.
- millora en la integració que no són Microsoft, gràcies a que permet una integració al sistema sense tenir acoblaments ajustadament.

### 1.2.3. Ontologia

S'entén per ontologia (des del punt de vista informàtic) com a la definició de conceptes existents en un domini, descrits segons les seves propietats, els seus tipus i les relacions que estableixen amb altres conceptes. Es podria entendre com un mapa conceptual detallat sobre un concepte.

La universitat de Stanford (Noy y McGuinness s.f.) defineix l'ontologia com la descripció formal i específica de conceptes en un domini, les propietats de cada concepte describint diferents atributs dels

conceptes i les restriccions. Una ontologia juntament amb un conjunt individual de instàncies de classes constitueix una base de coneixement.

Per tal d'exemplificar-ho, ells defineixen una classe anomenada vins, que representa tots els tipus de vins. Les instàncies d'aquesta classe podrien ser vins específics. A més, aquesta classe pot dividir-se en subclasses, com blanc, negre o rosat. I les possibles propietats dels mateixos poden ser tals com el cos, l'aroma o la quantitat de sucre que conté, així com qui n'és el principal productor.

És a dir, crear una ontologia implica diferents objectius:

- Definir les classes d'una ontologia
- Jerarquitzar les diferents classes en relacions de subclasse-superclasse
- Definir les propietats i quin tipus informació hi ha de figurar
- Omplir la informació referent a propietats i instàncies

#### **1.2.4. Estat de l'art**

L'ontologia és una tècnica que es va començar a utilitzar a partir dels anys 70 i que, amb el pas dels anys, gràcies a aquesta capacitat per generar bases de coneixement, ha anat guanyant cada vegada més força, fins al punt de generar la necessitat de crear conceptes com enginyer ontològic o enginyeria ontològica, que es basen en la creació i comprensió dels mapes conceptuals coneguts com a ontologia.

És gràcies a aquest impacte que tenen, que s'han desenvolupat diferents llenguatges pel funcionament d'ontologies, tals com KIF, OWL, RDF+RDF(S) o DAML+OIL, cadascun amb diferents finalitats, com ara descripció basada en la lògica, llenguatges basats en sintaxi tradicional.

Tal és la influència que tenen, que l'entorn de l'enginyeria també s'ha vist afectat, fins al punt que han sorgit diferents aplicacions que permeten treballar amb ontologies.

D'una banda, la universitat de Stanford ha desenvolupat una aplicació, anomenada Protégé, d'accés gratuït que permet organitzar la informació en superclasses i subclasses, així com crear instàncies, propietats i relacions. A més a més, ho complementen amb una guia sobre com crear una ontologia fent servir la seva aplicació

## 2. Introducció

### 2.1. Objectius del treball

- Dissenyar i desenvolupar un model ontològic genèric, aplicable a diferents tipus de problemes dins d'una mateixa classe general.
- Desenvolupar una simulació i optimització del cas a estudiar.
- Desenvolupar l'entorn de comunicacions adient per a englobar els diferents models en un de sol.
- Desenvolupar un prototipus que validi la viabilitat del projecte.

És a dir, l'objectiu d'aquest treball és generalitzar els problemes, de la mateixa manera que les diferents funcions lineals, com:

$$y = 3x + 4 \quad (\text{Eq.1})$$

$$y = -4x + 2 \quad (\text{Eq.2})$$

Es poden generalitzar en una sola funció:

$$y = ax + b \quad (\text{Eq.3})$$

### 2.2. Abast del treball

Arran dels objectius plantejats previament, s'ha definit quins elements s'han d'utilitzar i com s'han de comunicar entre ells.

D'una banda, cal que hi hagi un *Really Simple Syndication*, abreujat com a RSS, ja que proporciona dos funcions essencials per aquest tipus de casos:

- Permet compartir contingut en format XML de manera contínua, format molt útil en aquests casos donat que qualsevol programa o aplicació els pot llegir.
- Permet que qualsevol usuari pugui subscriure's i rebre aquest contingut XML.

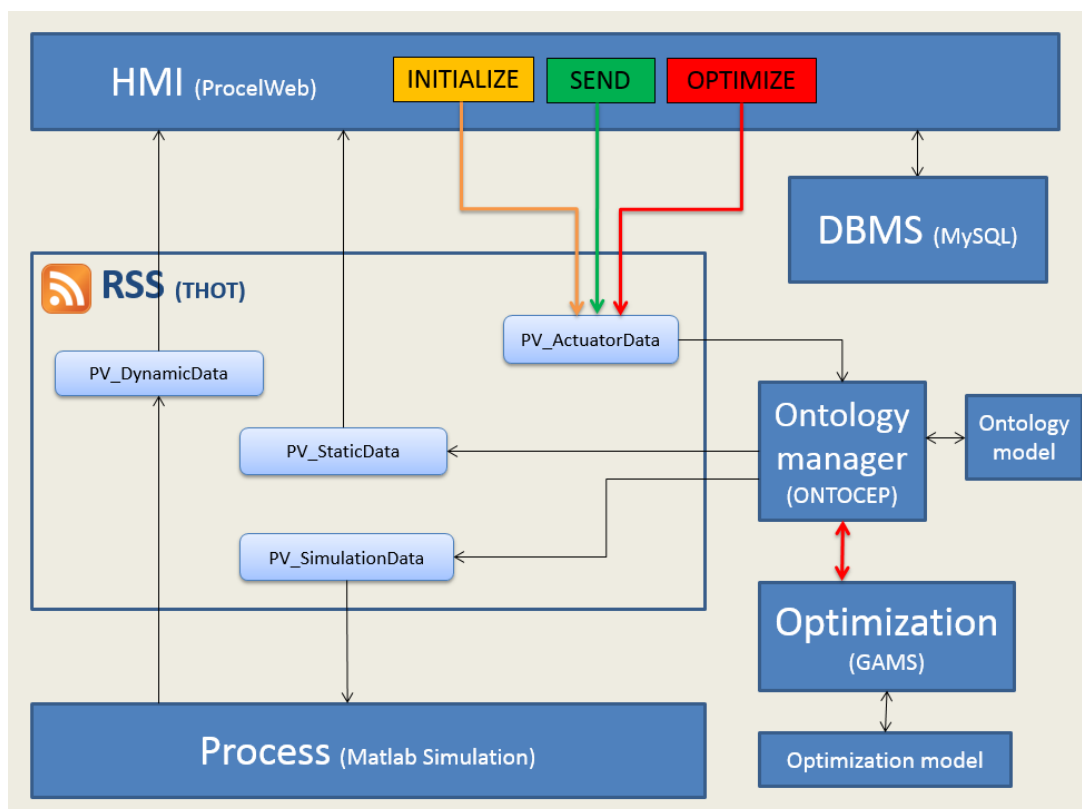
Degut a aquestes dues funcions, un RSS permet controlar amb relativa facilitat el flux d'informació, actuant com a nucli del sistema, sense el qual, seria molt més difícil treballar.

D'altra banda, cal saber qui o què envia una informació i qui l'ha de rebre. En el cas d'estudi d'aquest treball, són tres elements els qui s'han de comunicar entre ells:

- Simulació, que és l'encarregada de reproduir la realitat del problema.
- Optimització, que és qui determina com s'han d'organitzar els diferents elements del problema d'acord a una funció objectiu.
- Ontologia, que serveix com a eina d'organització de la informació així com de traductor entre l'ontologia i la simulació i la optimització.

Aquests 3 elements poden anar soportats per altres aplicacions que facilitin el seu funcionament, com ara una interfície d'usuari, que faciliti la gestió de la informació.

L'esquema que exemplifica la unió de tots aquests elements es mostra a la figura 6.



**Figura 6.** Esquema del bucle de control.

Per tant, per fer funcionar aquest bucle, primer cal encendre el RSS i configurar la IP segons com es treballi, doncs les altres aplicacions buscaran el RSS per poder connectar-s'hi i obtenir els arxius que necessitin.

En cas d'executar totes les aplicacions des del mateix ordinador, la IP que es farà servir serà *localhost*. En cas contrari, si les aplicacions s'executen des d'ordinadors diferents, la IP que interessarà serà la de l'ordinador on s'estigui executant el RSS.

El següent pas és encendre la simulació del procés i, juntament amb ella, l'aplicació que permetrà enviar i rebre els arxius que la simulació requereixi, sempre que hagin estat definits a la mateixa aplicació.

El tercer pas és encendre la HMI i connectar-la només al RSS, deixant-la en espera. Després d'això, cal encendre el gestor de l'ontologia i connectar-la al RSS. Finalment, cal tornar a la HMI i donar l'ordre de inicialitzar, que permetrà generar els arxius xml amb les dades inicials registrades a l'ontologia.





## **3. Presa de decisions**

### **3.1. Suport ontològic**

D'entre les diferents eines que es poden trobar que permetin desenvolupar ontologies, es va optar per Protégé donat que era intuïtiva i fàcil de manipular, la qual permetia revisar l'ontologia per detectar errors i corregir els que poguessin aparèixer sense dificultat.

### **3.2. Simulador emprat**

D'entre les diferents eines que es poden trobar al mercat per a simular processos, s'ha optat per l'ús del MATLAB ja que és una altra eina utilitzada com a simulador de processos quan treballa juntament amb simulink. La decisió de MATLAB rau en el fet que és, actualment, una de les eines matemàtiques més utilitzades, gràcies a la varietat de funcions de les quals disposa i que permet recrear la gran majoria de problemes.



## 4. Desenvolupament

Tot el projecte s'ha desenvolupat entorn a l'optimització sobre la cadena de subministrament elèctric d'una casa mitjançant energies renovables, bateries i connexió a la xarxa (Silvente, et al 2012).

### 4.1. Optimització

Per tant, el primer pas del projecte va ser estudiar l'optimització i analitzar quina informació rebria la simulació des de l'optimització, quins paràmetres i sets eren els que s'havien d'incloure a l'ontologia i quins d'aquests paràmetres podien ser de caràcter dinàmic, per tal de saber la referència ontològica per guardar-ne els nous valors i, en executar de nou l'optimització, utilitzar els nous valors. Apart, calia que aquesta informació estigués emmagatzemada tant a l'ontologia com en arxiu de text, principalment txt, doncs és la manera més senzilla de poder canviar les dades en un arxiu de GAMS.

Els sets que s'havien d'incloure a l'ontologia eren:

- Fonts d'energia
- Consumidors d'energia
- Sistemes d'emmagatzematge energètic
- Intervals de temps
- Repeticions

D'altra banda, els paràmetres que s'havien de contemplar eren:

- Consum de cada mercat per diferents repeticions en kWh i la seva duració en intervals de 15 min
- La màxima energia que es pot subministrar per cada una de les fonts, així com la màxima i mínima quantitat d'energia que pot emmagatzemar-se, tot en kWh
- Costos de producció i emmagatzematge en €/kWh i insatisfacció de demanda en €/15 min
- Mínim temps inicial i màxim temps final per cada consum en intervals de 15 min

De tots aquests paràmetres, el que principalment havia de ser dinàmic era la màxima producció d'energia. Donat que al simulació havia d'utilitzar dades de producció energètica reals, es va decidir que el valor que fes servir la simulació s'hauria de sobreesciure en els valors de l'optimització, per tal d'adequar-se millor a allò que es volia representar.

Finalment, les variables que interesaven per la simulació eren:

- Energia subministrada per cada font a cada interval de temps, en kWh
- Energia subministrada i rebuda pel sistema d'emmagatzematge, en kWh
- Temps d'inici de cada consum, en intervals de 15 min

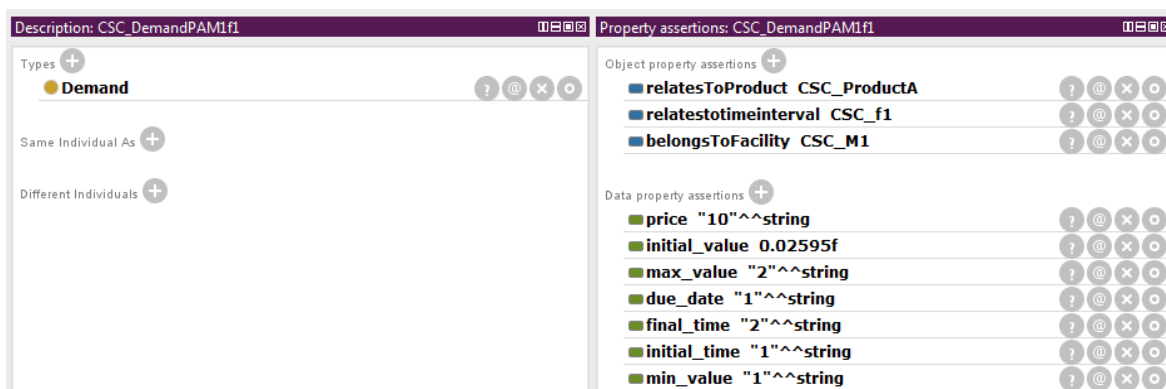
Per tal que funcionés, el bucle havia d'executar l'optimització no des de l'arxiu de GAMS directament, sinó des d'un arxiu de projecte en GAMS.

## 4.2. Ontologia

Sabent quina informació s'havia d'incloure de l'optimització, es va procedir a la construcció de la mateixa. Per això, es van fer servir dues eines: OntoCEP, desenvolupada pel grup de recerca CEPIMA, i Protégé, desenvolupada per la universitat de Stanford com a software gratuït. La primera aplicació serveix per crear tots les instàncies necessàries per a l'ontologia, mentre que la segona serveix per poder corregir-la en cas d'errors o per poder veure si les instàncies s'han creat correctament.

El primer que cal saber és que l'ontologia funciona en base a classes, instàncies individuals de cada classe, propietats de les instàncies i relacions de dependència amb altres instàncies. Des del punt de vista ontològic, la classe és l'instància que volem representar, les propietats són les característiques, generalment de caire numèric, que defineixen la classe i les relacions són vincles que permeten delimitar la instància d'entre els mateixos de la mateixa classe.

Per exemple, una instància pot ser la demanda del producte A (PA), del mercat 1 (M1), la primera vegada que aquest requereix producte A (f1), està ja definida i es vol veure la informació des de Protégé.



**Figura 7.** Exemple de les instàncies i la informació que poden contenir.

D'una banda, a la franja de color morat apareix el nom de CSC\_DemandPAM1f1, que seria l'identificador de la instància que es vol definir. El nom és important que segueixi un patró, doncs facilita la identificació i la cerca dels instàncies.

D'altra banda, es poden distingir 3 blocs amb informació.

- El primer és el que rep el nom de *Types*, que ens està indicant a quina classe pertany la instància CSC\_DemandPAM1f1. En aquest cas, pertany a la classe Demanda.
- El segon rep el nom de Object property assertions, que corresponen a les relacions que té la instància amb instàncies d'altres classes i que delimiten la pròpia instància. Tal com s'observa a la figura, la instància CSC\_DemandPAM1f1 està relacionad amb el producte A i a la infraestructura M1 durant la repetició f1.
- El tercer bloc s'anomena Data property assertions, que inclou la informació referent als valors que pot tenir dita instància. Per exemple, donat que és una demanda, es poden definir propietats com el preu, el valor que es vol obtenir o a quin moment fa referència aquella instància.

A partir d'aquí, es van crear les diferents instàncies des d'OntoCEP, ja que permet crear un número n d'instàncies a partir d'un patró. Primer, cal determina a quina classe pertany la instància que volem crear i, després es defineix el patró a seguir. Aquest patró, prenent l'exemple de la demanda, pot ser CSC\_DemandPAM1f, i es demana a OntoCEP que crei 96 instàncies.

Crear nuevas instancias : Nombre de las Instancias

**Previsualización**

Nombres

- CSC\_DemandPAM1f1
- CSC\_DemandPAM1f2
- CSC\_DemandPAM1f3
- CSC\_DemandPAM1f4
- CSC\_DemandPAM1f5
- CSC\_DemandPAM1f6
- CSC\_DemandPAM1f7
- CSC\_DemandPAM1f8
- CSC\_DemandPAM1f9
- CSC\_DemandPAM1f10
- CSC\_DemandPAM1f11
- CSC\_DemandPAM1f12
- CSC\_DemandPAM1f13
- CSC\_DemandPAM1f14
- CSC\_DemandPAM1f15
- CSC\_DemandPAM1f16
- CSC\_DemandPAM1f17
- CSC\_DemandPAM1f18
- CSC\_DemandPAM1f19
- CSC\_DemandPAM1f20
- CSC\_DemandPAM1f21
- CSC\_DemandPAM1f22
- CSC\_DemandPAM1f23
- CSC\_DemandPAM1f24
- CSC\_DemandPAM1f25
- CSC\_DemandPAM1f26
- CSC\_DemandPAM1f27
- CSC\_DemandPAM1f28

**Nombres Fijos**

☒ Crear Nombres desde un patrón

Patrón de Nombre

Número de Instancias

**Nombres Relacionales**

☐ Trabajar con Nombres Relacionales

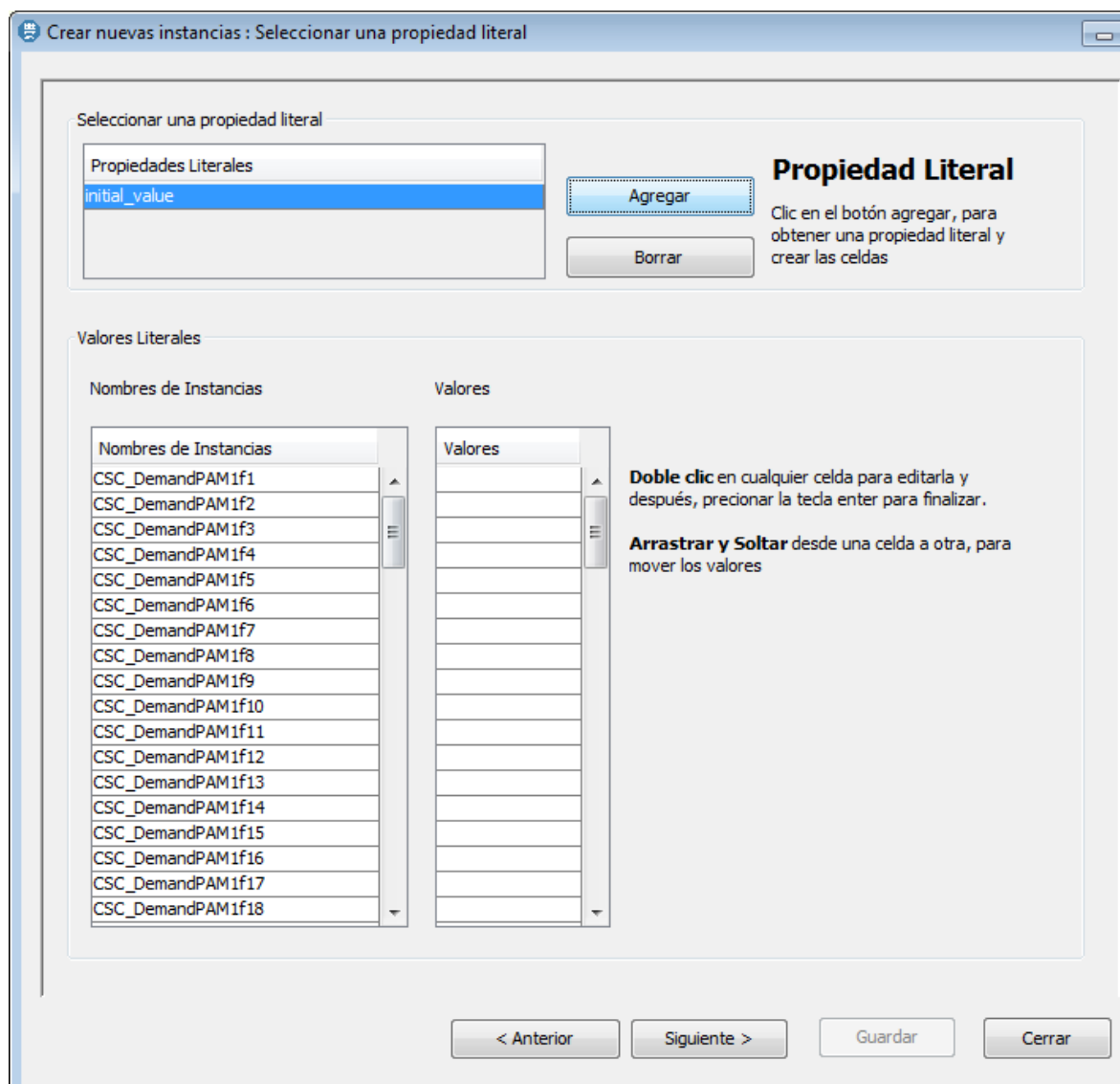
**Edición**

☒ Remplazar Caracter  Por

☐ Agregar Patrón

**Figura 8.** Exemple de com s'escriuen les instàncies a OntoCEP.

El següent pas és determinar una propietat de les Data property assertions i inserir un valor per a cada instància creada. Com per exemple `initial_value`.



**Figura 9.** Exemple de com s'afegeixen valors a cada instàncies des de OntoCEP.

El tercer i penúltim pas és definir els vincles amb altres instàncies seleccionant l'opció dins dels valors de l'ontologia i arrossegant al valor de l'objecte que es vol. Finalment, seleccionar l'opció de guardar.

Crear nuevas instancias : Propiedades de Objeto

Seleccione propiedades de Objetos

Propiedad de Objeto

belongsToFacility

Agregar

Borrar

**Seleccione una Propiedad de Objeto**

clic en el botón agregar para seleccionar una propiedad desde la Ontología

Valores de Objeto

Nombres de Instancias

Valores

Opciones

Nombres de Instancias

CSC\_DemandPAM1f1

CSC\_DemandPAM1f2

CSC\_DemandPAM1f3

CSC\_DemandPAM1f4

CSC\_DemandPAM1f5

CSC\_DemandPAM1f6

CSC\_DemandPAM1f7

CSC\_DemandPAM1f8

CSC\_DemandPAM1f9

CSC\_DemandPAM1f10

CSC\_DemandPAM1f11

CSC\_DemandPAM1f12

CSC\_DemandPAM1f13

CSC\_DemandPAM1f14

CSC\_DemandPAM1f15

Valores de Objeto

Valores de la Ontología

CSC\_M1

CSC\_M10

CSC\_M11

CSC\_M12

CSC\_M13

CSC\_M14

CSC\_M15

CSC\_M16

CSC\_M17

CSC\_M18

CSC\_M19

CSC\_M2

CSC\_M20

CSC\_M21

CSC\_M22

Arrastrar y Soltar valores desde la columna Opciones a la columna valores

Borrar

Obtener opciones desde la Ontol...

< Anterior

Siguiente >

Guardar

Cerrar

**Figura 10.** Exemple de com es creen les relacions de les instàncies.

En acabar, cada instància tindria una forma similar a la de la figura 7.

Finalment, cal construir el diccionari que permet traduir la instància i la propietat numèrica desitjada a un nom que sigui enteligible per la simulació i optimització. Per fer-ho, cada instància i la propietat que es vol traduir s'ha de transformar a algun nom/referència. Per exemple, a la figura 11 es mostra com el initial\_value de la demanda fa referència a CSC\_ProductA.CSC\_M10.CSC\_f1 al camp field amb el valor que partoca de initial\_value. Aquesta referència ha de ser única, així com el nom al qual es tradueixi, per evitar que hi hagi duplictat de la informació o eliminació d'aquesta.



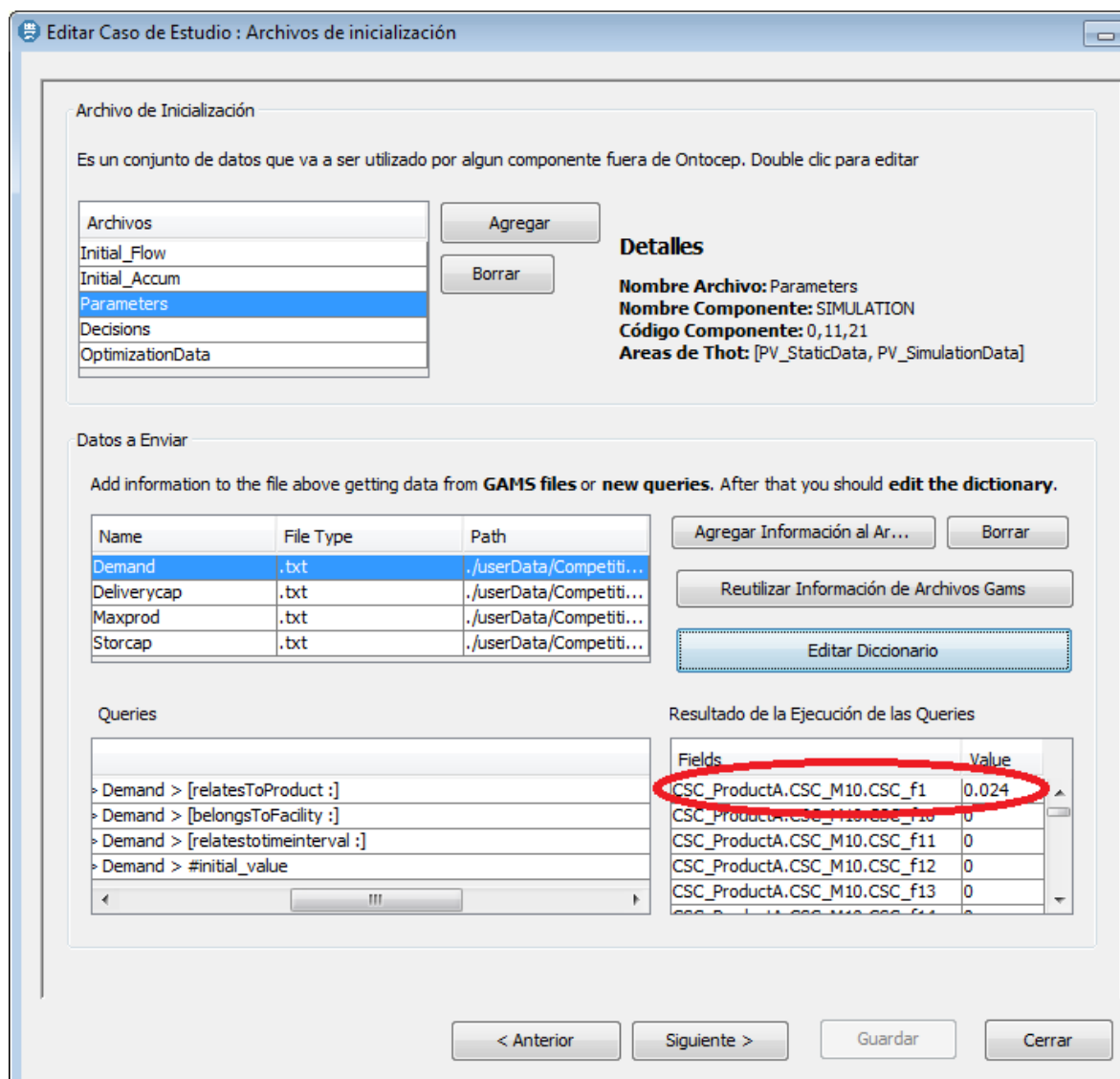


Figura 11. Exemple de referència pel diccionari.

A la vegada que es crea aquest diccionari, també es crea la informació que més endavant s'ha d'enviar a GAMS en format txt.

**Crear un Archivo de Salida**

Nombre Archivo:

Tipo Archivo:

Guardar en:

Categoría:

Camino a la Clase:

Query

ProductionProcess > ProcessManagement > ProductionRequirement > Demand > [relatesToProduct :]

ProductionProcess > ProcessManagement > ProductionRequirement > Demand > [belongsToFacility :]

ProductionProcess > ProcessManagement > ProductionRequirement > Demand > [relatestotimeinterval :]

ProductionProcess > ProcessManagement > ProductionRequirement > Demand > #initial\_value

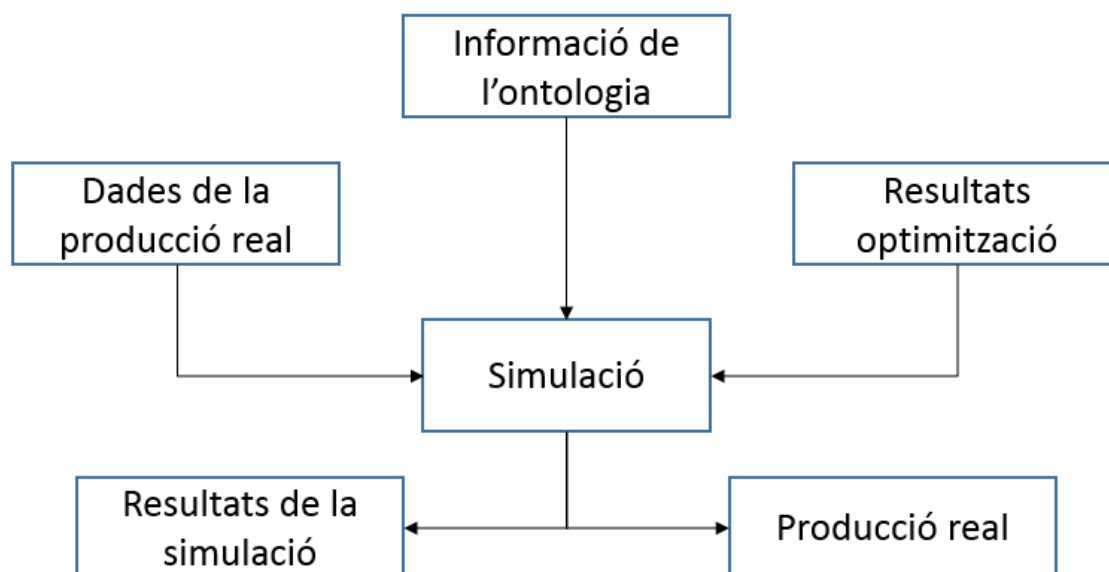
Fields	Value
CSC_ProductA.CSC_M10.CSC_f1	0.024
CSC_ProductA.CSC_M10.CSC_f10	0
CSC_ProductA.CSC_M10.CSC_f11	0
CSC_ProductA.CSC_M10.CSC_f12	0
CSC_ProductA.CSC_M10.CSC_f13	0
CSC_ProductA.CSC_M10.CSC_f14	0
CSC_ProductA.CSC_M10.CSC_f15	0
CSC_ProductA.CSC_M10.CSC_f16	0
CSC_ProductA.CSC_M10.CSC_f17	0
CSC_ProductA.CSC_M10.CSC_f18	0
CSC_ProductA.CSC_M10.CSC_f19	0
CSC_ProductA.CSC_M10.CSC_f2	0.024
CSC_ProductA.CSC_M10.CSC_f20	0

**Figura 12.** Exemple de la creació individual dels arxius.

A partir d'aquí, l'ontologia encara pot patir canvis si es veu que la informació no es guarda on s'hauria de guardar o que els valors que s'envien són erronis, però es té una primera versió de l'ontologia amb la qual es pot treballar adequadament.

### 4.3. Simulació

Finalment, amb l'ontologia ja creada i sabent quina informació havia de llegir i enviar la simulació, es va procedir al desenvolupament de la mateixa. De manera esquemàtica, la simulació havia d'organitzar la informació de la següent manera.



**Figura 13.** Diagrama de flux de la informació de la simulació.

D'una banda, requeria de 3 inputs d'informació:

- Dades de la producció reals, per a poder realitzar el balanç d'energia i calcular si s'ha d'utilitzar energia de les bateries i, en cas de no poder subministrar energia, agafar energia de xarxa.
- Informació de l'ontologia referent al nom de les instàncies a quina propietat fan referència, per poder llegir la informació i, posteriorment, guardar-la.
- Resultats de l'optimització per poder comparar si allò que s'ha volgut optimitzar és correcte o no i treballar en base als temps inicials de cada tasca determinats per l'optimització.

D'altra banda, es necessitaven 2 outputs:

- Els resultats de la simulació que s'havien de guardar a l'ontologia.
- Les dades de producció real conegudes per tal de poder utilitzar-les a l'optimització.

La simulació parteix de diferents arxius xml que contenen informació referent a:

- Interval de temps total de la simulació i l'increment de temps entre cada repetició de la simulació.
- Els valors inicials de les variables, així com el número de mercats, productes, productors i períodes de temps.

- El nom de les instàncies de cada una de les variables.

Per tant, en base a tota aquesta informació, el primer que cal fer és calcular la quantitat d'energia que es produeix durant aquell interval de temps, d'acord a les dues equacions següents:

$$P_{aire} = \frac{1}{2} \cdot A \cdot \rho_{aire} \cdot v_{aire}^3 \quad (\text{Eq.4})$$

$$P_{solar} = A \cdot i \quad (\text{Eq.5})$$

Un cop coneguda quan es produeix de cada energia, cal comprobar el balanç d'energia

$$P_{aire} + P_{solar} = Consum \quad (\text{Eq.6})$$

Si aquest valor és superior a 0, voldrà dir que tenim un excés d'energia i que, per tant, l'excedent s'hauria d'emmagatzemar. Si, en cas contrari, el valor fos inferior a 0, significaria que l'energia generada no és suficient i caldria veure si l'energia de les bateries és suficient per satisfer la demanda restant.

$$Bateria + P_{aire} + P_{solar} = Consum \quad (\text{Eq.7})$$

De nou, si aquest valor fos positiu, implicaria que la bateria és suficient per acabar de satisfer la demanda. Però, si el valor fos negatiu, s'hauria de subministrar la quantitat d'energia restant des de la xarxa directament, cosa que implicaria un cost addicional.

$$P_{xarxa} = Consum - (Bateria + P_{aire} + P_{solar}) \quad (\text{Eq.8})$$

El cost total serà la suma del cost d'emmagatzemar, el cost de consumir i el cost de subministrar des de la xarxa.

Finalment, aquests resultats s'han d'enviar a l'ontologia de nou per ser guardats<sup>1</sup>.

---

<sup>1</sup> Tota la informació referent a Matlab pot trobar-se a l'Annex.

## 5. Resultats

En aquest apartat es mostraran els resultats obtinguts que no requereixen una demostració del funcionament, donada la dificultat de plasmar una demostració de manera escrita.

A més, no s'inclourà el resultat de la simulació, que és el codi en sí, ja que es pot trobar a l'Annex adjunt.

### 5.1. Optimització

L'optimització s'ha adaptat d'acord als requeriments del sistema, de tal manera que tant els sets com els paràmetres, en lloc d'estar introduïts directament a la interfície de GAMS, s'introdueixen mitjançant arxius .set i .txt respectivament.

```
Sets
i energy producers /i1*i14/
j energy consumers /j1*j31/
k energy storage systems /k1/
f repetitions /f1*f96/
t time intervals /t1*t97/

Sets
i energy producers
/
$include userData\Competitive_SC\outputFiles\Factory.set
/
j energy consumers
/
$include userData\Competitive_SC\outputFiles\Market.set
/
k energy storage systems
/
$include userData\Competitive_SC\outputFiles\DistributionCenter.set
/
f repetitions
/
$include userData\Competitive_SC\outputFiles\TaskRepetition.set
/
t time intervals
/
$include userData\Competitive_SC\outputFiles\TimeInterval.set
/
```

**Figura 14.** Comparativa entre els sets inicials del GAMS (sets superiors) i els nous sets del GAMS (sets inferiors).



I, a més a més, la capacitat per generar diferents arxius en format xml, txt i sets que serveixen per fer funcionar tot el bucle.

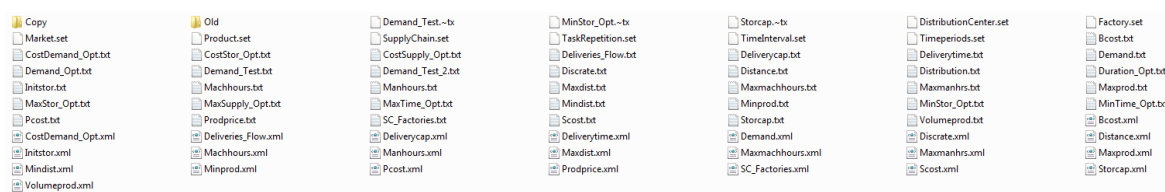


Figura 17. Llistat d'arxius creats des de l'ontologia.

## 5.3. Documentació

Apart dels resultats com a tals de l'aplicació de les aplicacions, també s'obté un resultat en forma de coneixement, que permet millorar els prototips que es puguin desenvolupar en un futur, que cal documentar segons siguin observacions de la implementació o de l'arquitectura.

### 5.3.1. Implementació

- Cal analitzar el problema en conjunt, és a dir, simulació i optimització simultàniament, per facilitar la creació de l'ontologia. En cas de treballar partint d'una simulació o optimització, el més recomanable és adaptar el que es desenvolupi a allò del que s'està partint.
- Reduir el màxim possible la quantitat d'instàncies a crear, donat que el pes augmenta ràpidament i això dificulta, d'una banda, la transferència de la informació i, de l'altra, la lectura dels documents per part de les aplicacions.

### 5.3.2. Arquitectura

- És recomanable, si és possible, reduir el nombre d'aplicacions obertes simultàniament a l'ordinador, doncs cada aplicació ocupa una memòria de l'ordinador i, si afegim el fet que ha d'anar processant informació amb Matlab o GAMS mentre en manté d'altres obertes, és relativament senzill que pugui colapsar.
- Cal tenir en compte que no serveix qualsevol ordre per executar les aplicacions, obrir una aplicació abans que l'altra pot provocar que després el sistema no funcioni degudament. Aquest ordre és: Thot – T.C.G – Matlab – ProcelWeb – OntoCEP – Ordre de ProcelWeb.





## Conclusions

Després de desenvolupar aquest projecte, les conclusions a les quals s'han arribat són:

- S'ha aconseguit dissenyar i desenvolupar un model ontològic genèric, ja que, com està definit en base a uns conceptes genèrics, permet que sigui fàcil d'adaptar dins els diferents casos de la cadena de subministrament que es poden trobar.
- S'ha desenvolupat una simulació que respon a les necessitats plantejades al llarg del document, tals com la necessitat de resoldre el balanç d'energia o de poder comunicar-se amb les altres aplicacions.
- S'ha aconseguit implementar el conjunt de tal manera que sigui capaç comunicar-se de manera adient i compartir la informació.
- S'ha demostrat mitjançant aquest prototipus la viabilitat d'aquesta tecnologia i que, mitjançant un esforç addicional i un desenvolupament de noves tecnologies i tècniques, pot servir de manera eficient per a que els futurs prototipus puguin plantejar i resoldre problemes.

Val a dir que, malgrat el resultat final sigui el funcionament del conjunt, la major part de l'esforç i les hores invertides és a nivell pràctic, en crear, corregir i detectar els errors que poguessin generar-se.



## Treball futur

De cara al futur, per tal de millorar aquest projecte, caldria:

- Validar el funcionament del mateix amb altres casos de cadena de subministrament més simples. Processos més complexos implicaria invertir temps en crear noves instàncies amb les respectives propietats.
- Millorar el rendiment de les aplicacions mantenint la quantitat d'informació que es fa servir però reduint el temps necessari per processar-la i manipular-la.
- Desenvolupar un nou cas d'estudi diferent a la cadena de subministrament, com per exemple el cas Tennessee Eastman, donat que és un cas conegut i d'estudi des del punt de vista de l'enginyeria química.



## Pressupost i/o Anàlisi Econòmica

### Costos de personal

S'han estimat al voltant de 600 hores de treball per part d'un eninyer químic, així com 50 de consuler, incloent un 33% de seguretat social.

**Taula 2.** *Costos de personal.*

Categoria	Hores	Cost unitari (sense SS)	Cost
Enginyer químic	600	12.5 €/h	7500 €
Consultor	50	15 €/h	750 €
Total (amb SS)			10972.5 €

### Costos de Software i Hardware

Els costos de de Software són els corresponents a MATLAB i Microsoft Office.

**Taula 3.** *Costos de Software.*

Categoria	Quantitat	Cost unitari	Cost
Matlab	1	1200 €/llicència	1200 €
Microsoft Office	1	540 €/llicència	540 €
Protégé		0 €/llicència	0 €
Total			1740 €

Altrament, els costos de Hardware són els de l'ordinador utilitzat amb la corresponent amortització de 5 anys, que és el moment en que es sol considerar que un ordinador està obsolet, en quant a components.

**Taula 4.** *Costos de Hardware.*

Categoria	Quantitat	Cost unitari	Amortització	Cost
Ordinador	1	1000 €/unitat	5 anys	200 €
Total				200 €

### Costos totals

Per tant, els costos totals seran la suma dels costos anteriors.

**Taula 5.** *Costos totals.*

Categoria	Cost
Costos de personal	10972.5 €
Costos de Software	1740 €
Costos de Hardware	200 €
Total	12912.5 €

## Bibliografia

Committee on Challenges for the Chemical Sciences in the 21st Century, National Research Council. 2003. *Chemical theory and computer modelling. In Beyond the molecular frontier. Challenges for Chemistry and Chemical Engineering*. Washington: The National Academies Press.

Institution of Chemical Engineers. n.d. *IChemE Process Management & Control Special Interest Group*. [http://www.icheme.org/communities/special-interest-groups/process%20management%20and%20control/about\\_us.aspx](http://www.icheme.org/communities/special-interest-groups/process%20management%20and%20control/about_us.aspx).

Noy, Natalya, y Deborah McGuinness. s.f. *Ontology Development 101: A Guide to Creating Your First Ontology*. Último acceso: 1 de 10 de 2016. [http://protege.stanford.edu/publications/ontology\\_development/ontology101-noy-mcguinness.html](http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html).

Silvente, Javier, Moisès Graells, Antonio Espuña, y Pep Salas. 2012. *An optimization model for the management of energy supply and demand in smart grids*. IEEE. 424-429.

Skeie, Nils-Olav. 2013. *Industrial Information Technology*. Capítols: 4, 22-23.

Venkatasubramanian, Venkat. 2009. *DROWNING IN DATA: Informatics and Modeling Challenges in a Data-Rich Networked World*. AIChE Journal.

## Bibliografia complementària

Kokossis, A.; Gao, E. i Kourakis, A. 2006. *On the dynamic management of chemical engineering knowledge using an ontology-based approach*. 16<sup>th</sup> European Symposium on Computer Aided Process Engineering.

Jurisica, I.; Mylopoulos, J. i Yu, E. *Using ontologies for Knowledge Management: An Information System Perspective*. Últim accés: 19 de juny de 2017. <http://ftp.cs.toronto.edu/pub/eric/asis99.pdf>

Ide, N. i Woolner, D. *Historical Ontologies*. Últim accés: 19 de juny de 2017. <http://www.cs.vassar.edu/~ide/papers/festschrift.pdf>







## Annex 1: Guia d'usuari

### A1. Arquitectura

Aquest projecte està format per 6 elements:

- Simulació
- Optimització
- Ontologia
- *Human Machine Interface* o HMI
- *Really Simple Syndication* o RSS (anomenat Thot)
- Aplicació de suport a la comunicació (T.G.C)

### A2. Seqüència d'inici

1. Executa el RSS, anomenat Thot.

Thot està a càrrec d'organitzar (en forma d'àrees) la informació que rep i envia als clients que s'han subscrit a les diferents àrees. Això vol dir que Thot ha de rebre i ensenyar la informació de cada un dels fitxers xml que puguin demanar o enviar altres elements.

Per tal de configurar Thot, cal:

- Copiar la carpeta del fitxer que conté Thot a l'ordinador on s'obrirà.
- Aconseguir la seva adreça IP (IPv4 address).
  - Cerca, a l'ordinador, cmd i escriu ipconfig
- Escriu la adreça IP a l'arxiu thot.config, a la mateixa carpeta que l'aplicació, thot.jar.

Aquest ordinador l'anomenarem Ordinador Principal

2. Obra el Matlab.

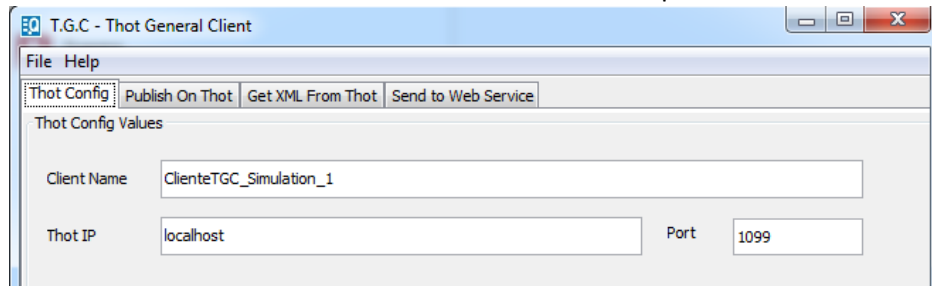
- Copia la carpeta del fitxer que conté la simulació a l'ordinador Client\_01, pot ser el mateix ordinador que l'ordinador on està corrent Thot.
- Executa la simulació anomenada SIM\_MAIN dins la mateixa carpeta Simulacio\_NOBADIS\_v2.

3. Connecta T.G.C a Thot

Per tal de configurar la connexió al RSS (Thot), cal:



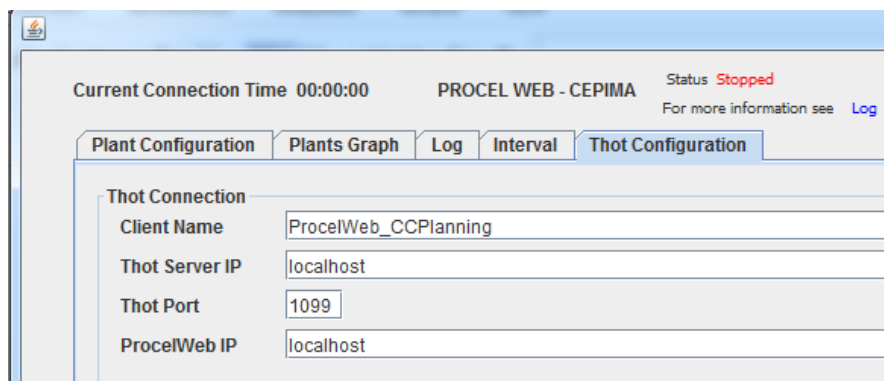
- Copiar la carpeta que conté T.G.C a l'ordinador Client\_01
- Configuració de T.G.C: un cop s'ha encès, s'ha d'obrir la pestanya Thot Config i introduir la IP de l'Ordinador Principal a l'opció de Thot IP.
  - Si l'Ordinador Principal i l'ordinador Client\_01 són el mateix, la adreça IP pot ser tant "localhost" com la IP de l'Ordinador Principal.



- Inicialitzar Thot: al menú general de T.G.C, obre la pestanya "Get XML From Thot" i especifica el path on es vulguin guardar els xml que es generaran.
- A la pestanya "Publish on Thot", especifica el path de la informació de la simulació que vulgui enviar-se a Thot.
- En aquest projecte,
  - L'arxiu que es llegeix des de Thot és PV\_SimulationData
  - L'arxiu que s'envia a Thot és PV\_DynamicData
  - Tots dos arxius són tractats per la simulació del procés.

#### 4. Obra la HMI, anomenada ProcelWeb.

- Copia la carpeta que conté els arxius de ProcelWeb a l'ordinador Client\_02
- Configura la HMI: al menú general de la HMI, obra la pestanya "Thot config" i introdueix la IP de l'Ordinador Principal a l'opció Thot Server IP.
  - Si l'Ordinador Principal i l'ordinador Client\_02 són el mateix, la adreça IP pot ser tant "localhost" com la IP de l'ordinador.
- Clica el botó Start.



#### 5. Engega OntoCEP

- Copia la carpeta que conté els arxius a l'ordinador Client\_03.
  - Configura OntoCep: la informació sobre la seva configuració es troba a l'arxiu txt anomenat MegaConfig.
    - Si no passa a l'estat de connectat automàticament, l'usuari ha d'anar a "Case Study, Thot" i Connect to Thot.
    -
6. L'usuari ha de tornar a Matlab i iniciar la simulació.
  7. Finalment, l'usuari ha de tornar a la HMI (ProcelWeb) i donar l'ordre d'inicialitzar. Amb això, el bucle començarà.

### A3. Seqüència d'Operació

En cas de tenir un problema en el funcionament o voler saber si està funcionant, la seqüència d'operació consta de 3 etapes:

- Introduir un canvi.
- Buscar una solució.
- Implementar una solució.

El canvi més eficaç sol ser un canvi de valor a una variable a un valor fàcil d'identificar, diferent dels valors de les altres variables dins la mateixa categoria. Per exemple, canviar el valor d'una demanda. Si tots els valors estan dins d'un rang determinat, introduir un valor fora del rang o un valor dins del rang fàcil d'identificar. Aquest canvi de valor deixarà un "rastre" que es podrà seguir mitjançant els diferents arxius.



## Annex 2: Codi Matlab

```
function SIM_MAIN
Filename='./Input/Input_Temporal/TemporalFrame.xml';
[t_period,t_paso,t_wait]=LeeTemporalFrame(Filename);
Filename='./Input/Input_Temporal/PV_StaticData.xml';
[DistCap,Nombre_VelDist_Tag,...
Demanda,nombre_dmd_tag,...
Distancia,nombre_dst_tag,...
PlantCap,nombre_pltcap_tag,...
StorCap,nombre_str_cap_tag,...
produccion,nombre_prod_tag,...
Distribucion,nombre_dist_tag,...
inicio,nombre_ini_tag,...
duracion,nombre_dur_tag,...
ProductNames,MarketNames,PeriodNames,PlantNames]=LeeStaticData(Filename)
;
Filename='./Input/Input_Temporal/PV_DynamicData.xml';
[IniProd,Nombre_IniProd_Tag,...
IniStor,nombre_IniStor_tag,...
IniDistr,nombre_IniDistr_tag,...
IniDel,nombre_IniDel_tag,...
IniProdSpeed,nombre_IniProdSpeed_tag,...
IniStorSpeed,nombre_IniStorSpeed_tag,...
IniDistrSpeed,nombre_IniDistrSpeed_tag,...
IniDelSpeed,nombre_IniDelSpeed_tag,...
ProductNames,MarketNames,TimeNames,PlantNames]=LeeDynamicData(Filename)
;
Filename='./Input/Input_Temporal/PV_MaxProdData.xml';
[nombre_maxprod_tag,maxprod_data,MaxProdPeriodNames]=LeeMaxProdData(Filename);
Filename='./Input/Input_Temporal/ExternalTarget.xml';
[Externo,ProductNames,PlantNames,nombre_Extr_tag]=LeeProduccion(Filename)
;
[FOBJ,Storage_acumulada,Produccion_acumulada,Delivery_acumulada]=simula(t_period,t_paso,t_wait,...
produccion,Distribucion,inicio,duracion,
Externo,Nombre_IniProd_Tag,nombre_IniProdSpeed_tag,nombre_IniStor_tag,...
nombre_IniStorSpeed_tag,nombre_IniDistr_tag,nombre_IniDistrSpeed_tag,nombre_IniDelSpeed_tag,...
nombre_IniDel_tag,PlantCap,DistCap,StorCap,Distancia,Demanda,IniStor,IniProd,IniDel,IniDistr,...
MarketNames,PlantNames,ProductNames,PeriodNames,Nombre_VelDist_Tag,nombre_dmd_tag,nombre_dst_tag,...
nombre_pltcap_tag,nombre_str_cap_tag,nombre_prod_tag,nombre_dist_tag,nombre_ini_tag,nombre_dur_tag,IniProdSpeed,IniStorSpeed,...
IniDistrSpeed,IniDelSpeed,TimeNames,nombre_maxprod_tag,maxprod_data,MaxProdPeriodNames);
end
```

```

function [t_period,t_paso,t_wait]=LeeTemporalFrame(Filename)
Main_tag='RecordElement';
%1.- Creación de la Especificación
estructura =struct('UTC_ini',{'Empty'},...
'UTC_fin',{'Empty'});
Componentes=fieldnames(estructura);
%2.- Lectura de una estructura intermedia general de último nivel
% (cualquier conjunto de campos, devolviendo texto)
estructura = Lee_XML(Filename,Main_tag,Componentes);
%3.- Traducción de acuerdo con la norma "Procelweb": Elemento,
TimeStasmp, Valor
dimen=size(estructura);
for elemento=1:dimen(2);
t1=char(estructura(elemento).(char(Componentes(1))));
t1(11:11)=' ';
period(elemento,1) = datenum(t1(1:23)) +
str2double(t1(24:26))/24;%columna 1
t1=char(estructura(elemento).(char(Componentes(2))));
t1(11:11)=' ';
period(elemento,2) = datenum(t1(1:23)) + str2double(t1(24:26))/24;
%columna 2
end
t_period=sort(period,1);
Nperiods=size(t_period);
for Period=2:Nperiods(1)
if(t_period(Period,1)>t_period(Period-1,2)+.001)
msgbox('error en los periodos');
end
if(t_period(Period,1)<t_period(Period-1,2)-0.001)
msgbox('warning en los periodos');
end
end
if Nperiods>1
t_paso=(t_period(2,2)-t_period(2,1))/(365*24*3600/5);
%t_paso es de 5 segundos
t_wait=t_paso/1;
end

```

```

function estruct_alfa= Lee_XML(filename, tagPrincipal, tagComponentes)
% Lectura de una estructura intermedia general de último nivel
% De los nodos con "tag principal" se seleccionan las propiedades
% indicadas por "tagComponentes", devolviendo texto)
%
% 1.- Cargar el documento
%
xDoc=xmlread(filename);
% 2.- Coger todos los nodos con el tag que nos interesa
Nodos = xDoc.getElementsByTagName(tagPrincipal);
% 3.- Preparar las estructuras: Cuántos nodos hay y cuántos elementos
% queremos escoger de cada nodo
numNodos = Nodos.getLength;
[numValores] = size(tagComponentes);
estruct_alfa = [];
% 4.- Lectura propiamente dicha
for nodo = 1 : numNodos
% 4.1. Situar en un nodo
Individuo = Nodos.item(nodo-1);
% 4.1.1. Situar en un elemento de interés
for valor=1:numValores(1)
elemento = Individuo.getElementsByTagName(tagComponentes(valor));
numero_de_IDS=elemento.getLength;
% 4.1.2. Comprobar que ese elemento es único
if (numero_de_IDS >1)
msgbox('error')
elseif(numero_de_IDS <0)
msgbox('error')
elseif(numero_de_IDS ==0)
% No hay este campo
string='zzz_empty';
% determinar su dimension
[dimen]=size(string);
% escribirlo
estruct_alfa=setfield(estruct_alfa,{nodo},....
tagComponentes{valor}, {1,1:dimen(2)},...
string);
else
% 4.1.3. Leerlo como string
string=char(elemento.item(0).getTextContent);
% determinar su dimension
[dimen]=size(string);
% escribirlo
estruct_alfa=setfield(estruct_alfa,{nodo},....
tagComponentes{valor}, {1,1:dimen(2)},...
string);
end
end
end
end

```

```

function [DistCap,Nombre_VelDist_Tag,...
demanda,nombre_dmd_tag,...
Distancia,nombre_dst_tag,...
PlantCap,nombre_pltcap_tag,...
StorCap,nombre_str_cap_tag,...
produccion,nombre_prod_tag,...
Distribucion,nombre_dist_tag,...
inicio,nombre_ini_tag,...
duracion,nombre_dur_tag,...
ProductNames,MarketNames,TimeNames,PlantNames]=LeeStaticData (Filename)
Main_tag='RecordElement';
estructura =struct('RecordElementID',{'Empty'},...
'UTC',{'Empty'},...
'NewValue',{'Empty'},...
'Category',{'Empty'},...
'Market',{'Empty'},...
'Site',{'Empty'},...
'EndProduct',{'Empty'},...
'TimeInterval',{'Empty'});
Componentes=fieldnames(estructura);
%2.- Lectura de una estructura intermedia general de último nivel
% (cualquier conjunto de campos, devolviendo texto)
estructura = Lee_XML(Filename,Main_tag,Componentes);
%3.- Traducción de acuerdo con la norma "Procelweb": Elemento,
TimeStasmp, Valor
dimen=size(estructura);
nombres=cell(4,dimn(2));
for elemento=1:dimn(2);
nombres(1,elemento)=cellstr(estructura(elemento).(char(Componentes(5))));
%Market
nombres(2,elemento)=cellstr(estructura(elemento).(char(Componentes(7))));
%EndProduct
nombres(3,elemento)=cellstr(estructura(elemento).(char(Componentes(8))));
%TimeInterval
nombres(4,elemento)=cellstr(estructura(elemento).(char(Componentes(6))));
%Site
end
for identific=1:4
Nnames=unique(nombres(identific,:));
[dummy,Nvalues]=size(Nnames);
if(strcmp(Nnames(Nvalues),'zzz_empty'))
Nnames=Nnames(1:Nvalues-1);
end
switch identific
case 1
MarketNames=Nnames;
[dummy,NMarkets]=size(MarketNames);
case 2
ProductNames=Nnames;
[dummy,NProducts]=size(ProductNames);
case 3
TimeNames=Nnames;
[dummy,NPeriods]=size(TimeNames);
case 4
PlantNames=Nnames;
[dummy,NPlantas]=size(PlantNames);
otherwise
end

```



```

end
DistCap=zeros (NProducts,NPlantas,NMarkets);
demanda=zeros (NProducts,NMarkets,NPeriods);
Distancia=zeros (NPlantas,NMarkets);
PlantCap=zeros (NProducts,NPlantas);
StorCap=zeros (NProducts,NPlantas);
produccion=zeros (NProducts,NPlantas,NPeriods);
Distribucion=zeros (NProducts,NPlantas,NMarkets,NPeriods);
for elemento=1:dimen(2)
Product=strmatch (cellstr (estructura (elemento) . (char (Componentes (7)))) ,Pro
ductNames, 'exact');
Market=strmatch (cellstr (estructura (elemento) . (char (Componentes (5)))) ,Mark
etNames, 'exact');
Period=strmatch (cellstr (estructura (elemento) . (char (Componentes (8)))) ,Time
Names, 'exact');
Planta=strmatch (cellstr (estructura (elemento) . (char (Componentes (6)))) ,Plan
tNames, 'exact');
valor=str2double (estructura (elemento) . (char (Componentes (3)))) ;
tag=cellstr (estructura (elemento) . (char (Componentes (1)))) ;
switch estructura (elemento) . (char (Componentes (4)))
case 'TransportResource.max_value'
DistCap (Product,Planta,Market)=valor;
Nombre_VelDist_Tag (Product,Planta,Market)=tag;
case 'Demand.value'
demanda (Product,Market,Period)=valor;
nombre_dmd_tag (Product,Market,Period)=tag;
case 'Distance.value'
Distancia (Planta,Market)=valor;
nombre_dst_tag (Planta,Market)=tag;
case 'LocationCapacity.max_value'
PlantCap (Product,Planta)=valor;
nombre_pltcap_tag (Product,Planta)=tag;
case 'InventoryParameter.max_value'
StorCap (:,Planta)=valor;
nombre_str_cap_tag (:,Planta)=tag;
case 'ProductionOrder.target_value'
produccion (Product,Planta,Period)=valor;
nombre_prod_tag (Product,Planta,Period)=tag;
case 'InventoryEndProduct.target_value'
case 'P_TransferRate.target_value'
Distribucion (Product,Planta,Market,Period)=valor;
nombre_dist_tag (Product,Planta,Market,Period)=tag;
case 'SupplyChainCost.target_value'
case 'Demand.min_value'
inicio (Product,Market,Period)=valor;
nombre_ini_tag (Product,Market,Period)=tag;
case 'Demand.due_date'
duracion (Product,Market,Period)=valor;
nombre_dur_tag (Product,Market,Period)=tag;
otherwise
msgbox ('error')
end
end

```

```

function [IniProd,Nombre_IniProd_Tag,...
IniStor,nombre_IniStor_tag,...
IniDistr,nombre_IniDistr_tag,...
IniDel,nombre_IniDel_tag,...
IniProdSpeed,nombre_IniProdSpeed_tag,...
IniStorSpeed,nombre_IniStorSpeed_tag,...
IniDistrSpeed,nombre_IniDistrSpeed_tag,...
IniDelSpeed,nombre_IniDelSpeed_tag,...
ProductNames,...
MarketNames,TimeNames,PlantNames]=LeeDynamicData (Filename)
Main_tag='RecordElement';
estructura =struct('RecordElementID',{'Empty'},...
'UTC',{'Empty'},...
'NewValue',{'Empty'},...
'Category',{'Empty'},...
'Market',{'Empty'},...
'Site',{'Empty'},...
'EndProduct',{'Empty'},...
'TimeInterval',{'Empty'});
Componentes=fieldnames(estructura);
%2.- Lectura de una estructura intermedia general de último nivel
% (cualquier conjunto de campos, devolviendo texto)
estructura = Lee_XML(Filename,Main_tag,Componentes);
%3.- Traducción de acuerdo con la norma "Procelweb": Elemento,
TimeStasmp, Valor
dimen=size(estructura);
nombres=cell(4,dimn(2));
for elemento=1:dimn(2);
nombres(1,elemento)=cellstr(estructura(elemento).(char(Componentes(5))));
%Market
nombres(2,elemento)=cellstr(estructura(elemento).(char(Componentes(7))));
%EndProduct
nombres(3,elemento)=cellstr(estructura(elemento).(char(Componentes(8))));
%TimeInterval
nombres(4,elemento)=cellstr(estructura(elemento).(char(Componentes(6))));
%Site
end
for identific=1:4
Nnames=unique(nombres(identific,:));
[dummy,Nvalues]=size(Nnames);
if(strcmp(Nnames(Nvalues),'zzz_empty'))
Nnames=Nnames(1:Nvalues-1);
end
switch identific
case 1
MarketNames=Nnames;
[dummy,NMarkets]=size(MarketNames);
case 2
ProductNames=Nnames;
[dummy,NProducts]=size(ProductNames);
case 3
TimeNames=Nnames;
[dummy,NPeriods]=size(TimeNames);
case 4
PlantNames=Nnames;
[dummy,NPlantas]=size(PlantNames);
otherwise
end
end

```

```

end
DistCap=zeros (NProducts,NMarkets,NPlantas);
demanda=zeros (NProducts,NMarkets,NPeriods);
Distancia=zeros (NPlantas,NMarkets);
PlantCap=zeros (NProducts,NPlantas);
StorCap=zeros (NProducts,NPlantas);
produccion=zeros (NProducts,NPeriods,NPlantas);
Distribucion=zeros (NProducts,NMarkets,NPeriods,NPlantas);
for elemento=1:dimen(2)
Product=strmatch(cellstr(estructura(elemento).(char(Componentes(7)))) ,Pro
ductNames,'exact');
Market=strmatch(cellstr(estructura(elemento).(char(Componentes(5)))) ,Mark
etNames,'exact');
Period=strmatch(cellstr(estructura(elemento).(char(Componentes(8)))) ,Time
Names,'exact');
Planta=strmatch(cellstr(estructura(elemento).(char(Componentes(6)))) ,Plan
tNames,'exact');
valor=str2double(estructura(elemento).(char(Componentes(3)))) ;
tag=cellstr(estructura(elemento).(char(Componentes(1)))) ;
switch estructura(elemento).(char(Componentes(4)))
case 'ProductionOrder.value'
IniProd(Product,Planta,Period)=valor;
Nombre_IniProd_tag(Product,Planta,Period)=tag;
case 'InventoryEndProduct.value'
IniStor(Product,Planta,Period)=valor;
nombre_IniStor_tag(Product,Planta,Period)=tag;
case 'P_TransferRate.value'
IniDistr(Product,Planta,Market,Period)=valor;
nombre_IniDistr_tag(Product,Planta,Market,Period)=tag;
case 'ProcessOutputMeasure.value'
IniDel(Product,Market,Period)=valor;
nombre_IniDel_tag(Product,Market,Period)=tag;
case 'ProductionOrder.initial_value'
IniProdSpeed(Product,Planta,Period)=valor;
nombre_IniProdSpeed_tag(Product,Planta,Period)=tag;
case 'InventoryEndProduct.initial_value'
IniStorSpeed(Product,Planta,Period)=valor;
nombre_IniStorSpeed_tag(Product,Planta,Period)=tag;
case 'P_TransferRate.initial_value'
IniDistrSpeed(Product,Planta,Market,Period)=valor;
nombre_IniDistrSpeed_tag(Product,Planta,Market,Period)=tag;
case 'ProcessOutputMeasure.initial_value'
IniDelSpeed(Product,Market,Period)=valor;
nombre_IniDelSpeed_tag(Product,Market,Period)=tag;
otherwise
msgbox('error')
end
end
end

```

```

function [ Nombre_MaxProd_Tag,maxprod_data, TimeNames] = LeeMaxProdData(
Filename )
Main_tag='RecordElement';
estructura =struct('RecordElementID',{'Empty'},...
'UTC',{'Empty'},...
'NewValue',{'Empty'},...
'Category',{'Empty'},...
'Market',{'Empty'},...
'Site',{'Empty'},...
'EndProduct',{'Empty'},...
'TimeInterval',{'Empty'});
Componentes=fieldnames(estructura);
%2.- Lectura de una estructura intermedia general de último nivel
% (cualquier conjunto de campos, devolviendo texto)
estructura = Lee_XML(Filename,Main_tag,Componentes);
%3.- Traducción de acuerdo con la norma "Procelweb": Elemento,
TimeStasmp, Valor
dimen=size(estructura);
nombres=cell(4,dimn(2));
for elemento=1:dimn(2);
nombres(1,elemento)=cellstr(estructura(elemento).(char(Componentes(5))));
%Market
nombres(2,elemento)=cellstr(estructura(elemento).(char(Componentes(7))));
%EndProduct
nombres(3,elemento)=cellstr(estructura(elemento).(char(Componentes(8))));
%TimeInterval
nombres(4,elemento)=cellstr(estructura(elemento).(char(Componentes(6))));
%Site
end
for identific=1:4
Nnames=unique(nombres(identific,:));
[dummy,Nvalues]=size(Nnames);
if(strcmp(Nnames(Nvalues),'zzz_empty'))
Nnames=Nnames(1:Nvalues-1);
end
switch identific
case 1
MarketNames=Nnames;
[dummy,NMarkets]=size(MarketNames);
case 2
ProductNames=Nnames;
[dummy,NProducts]=size(ProductNames);
case 3
TimeNames=Nnames;
[dummy,NPeriods]=size(TimeNames);
case 4
PlantNames=Nnames;
[dummy,NPlantas]=size(PlantNames);
otherwise
end
end
for elemento=1:dimn(2)
Product=strmatch(cellstr(estructura(elemento).(char(Componentes(7)))),Pro
ductNames,'exact');
Market=strmatch(cellstr(estructura(elemento).(char(Componentes(5)))),Mark
etNames,'exact');
Period=strmatch(cellstr(estructura(elemento).(char(Componentes(8)))),Time
Names,'exact');

```

```
Planta=strmatch(cellstr(estructura(elemento).(char(Componentes(6)))) ,PlantNames, 'exact');
valor=str2double(estructura(elemento).(char(Componentes(3))));
tag=cellstr(estructura(elemento).(char(Componentes(1))));
switch estructura(elemento).(char(Componentes(4)))
case 'ProductionOrder.max_value'
Nombre_MaxProd_Tag(Product,Planta,Period)=tag;
maxprod_data(Product, Planta, Period)=valor;
otherwise
msgbox('error')
end
end
end
```

```
function
[produccion,ProductNames,PlantNames,PeriodNames,nombre_tag]=LeeProduccion
(Filename)
Main_tag='RecordElement';
%1.- Creación de la Especificación demanda: mercado, producto, tiempo,
Valor
produccion=[];
estructura =struct('RecordElementID',{'Empty'},...
'PlantID',{'Empty'},...
'ProductID',{'Empty'},...
'PeriodID',{'Empty'},...
'NewValue',{'Empty'});
Componentes=fieldnames(estructura);
%2.- Lectura de una estructura intermedia general de último nivel
% (cualquier conjunto de campos, devolviendo texto)
estructura = Lee_XML(Filename,Main_tag,Componentes);
%3.- Traducción de acuerdo con la norma "Procelweb": Elemento,
TimeStasmp, Valor
dimen=size(estructura);
for elemento=1:dimen(2);
nombres(1,elemento)=cellstr(estructura(elemento).(char(Componentes(2))));
nombres(2,elemento)=cellstr(estructura(elemento).(char(Componentes(3))));
nombres(3,elemento)=cellstr(estructura(elemento).(char(Componentes(4))));
end
PlantNames=unique(nombres(1,:));
NPlants=size(PlantNames);
ProductNames=unique(nombres(2,:));
NProdcts=size(ProductNames);
PeriodNames=unique(nombres(3,:));
NPeriods=size(PeriodNames);
for Planta=1:NPlants(2)
for Product=1 : NProdcts(2)
for Period=1 : NPeriods(2)
for elemento=1:dimen(2)
if( strcmp(nombres(1,elemento),PlantNames(Planta)) && ...
strcmp(nombres(2,elemento),ProductNames(Product)) && ...
strcmp(nombres(3,elemento),PeriodNames(Period)) )
produccion(Product,Planta,Period)=str2double(estructura(elemento).(char(C
omponentes(5))));
nombre_tag(Product,Planta,Period)=cellstr(estructura(elemento).(char(Comp
onentes(1))));
else
elemento=elemento+1;
end
end
end
end
end
end
```

```
function [Valor, Fecha, Nombre_Tag]=LeeProcelWeb(Filename)
Main_tag='RecordElement';
%1.- Creación de la Especificación ProcelWeb: Elemento, TimeStasmp,
Valor
estructura =struct('RecordElementID',{'Empty'},...
'UTC',{9999},...
'NewValue',{9999});
Componentes=fieldnames(estructura);
%2.- Lectura de una estructura intermedia general de último nivel
% (cualquier conjunto de campos, devolviendo texto)
estruct_alfa = Lee_XML(Filename,Main_tag,Componentes);
%3.- Traducción de acuerdo con la norma "Procelweb": Elemento,
TimeStasmp, Valor
dimen=size(estruct_alfa);
for elemento=1:dimen(2);
Valor(elemento)=str2double(estruct_alfa(elemento).(char(Componentes(3))))
;
t1=char(estruct_alfa(elemento).(char(Componentes(2)))));
t1(11:11)=' ';
Fecha(elemento)= datenum(t1(1:23)) +
str2double(t1(24:26))/24;
Nombre_Tag(elemento)=cellstr(estruct_alfa(elemento).(char(Componentes(1))
));
end
end
```

```

function
[FOBJ,Storage_acumulada,Produccion_acumulada,Delivery_acumulada]=simula(t
_period,t_paso,t_wait,...
Production_target,Distribution_target,initial_time, duration_time,
Ext_target,Nombre_ProdTot_Tag,Nombre_VelProd_Tag,Nombre_Str_Tag,...
Nombre_VelStr_Tag,Nombre_TotDistr_Tag,Nombre_VelDistr_Tag,Nombre_VelMkDis
t_Tag,Nombre_TotDist_Tag,...
PlantCap,DistCap,StorCap,Distancia,Demanda,IniStor,IniProd,IniDel,IniDist
ributed,MarketNames,...
PlantNames,ProductNames,PeriodNames,Nombre_VelDist_Tag,nombre_dmd_tag,nom
bre_dst_tag,nombre_pltcap_tag,...
nombre_str_cap_tag,nombre_prod_tag,nombre_dist_tag,nombre_ini_tag,nombre_
dur_tag,IniProdSpeed,IniStorSpeed,IniDistrSpeed,IniDelSpeed,TimeNames,nom
bre_maxprod_tag,maxprod_data,MaxProdPeriodNames)
Mensaje={'Sin Mensajes' 'Sin Mensajes' 'Sin Mensajes' 'Sin Mensajes'};
NPeriods1=size(t_period);
NPeriods2=size(PeriodNames);
NPeriods=min(NPeriods1(1),NPeriods2(2));
NMaxProdPeriod=size(MaxProdPeriodNames);
[libre,NPlantas]=size(PlantNames);
[libre,NProducts]=size(ProductNames);
[libre,NMarkets]=size(MarketNames);
tiempo=t_period(1,1);
FOBJ=0.;
contador=0;
Storage_acumulada=IniStor;
ProduccionAcumulada=IniProd;
DeliveryAcumulada=IniDel;
% Càlcul de les velocitats (FlowRate)
ProdT=0;
DemandaT=0;
StorageT=0;
time_stamp=0;
%Dada que cal introduir amb el temporalframe. De moment es suposa
d'uns
%5 minuts.
for Period=1:NPeriods
while tiempo<t_period(Period,2)
vProd(:,:)=min
((Production_target(:,:,Period)-
IniProd(:,:,Period))/(t_period(Period,2)-(t_period(Period,1))),...
PlantCap(:,:));
if (Period>1)
vDist(:,:,:)=min(Distribution_target(:,:,Period)/(t_period(Period,2)-
t_period(Period,1)),...
DistCap(:,:,:));
else
vDist(:,:,:)=min((Distribution_target(:,:,Period)-
IniDistributed(:,:,Period))/(t_period(Period,2)-t_period(Period,1)),...
DistCap(:,:,:));
end
vDel(:,:,:)= vDist(:,:,:);
vExt(:,:)=Ext_target(:,:,Period)/(t_period(Period,2)-
t_period(Period,1));
disp(Mensaje);
time=datevec(tiempo);
[ YearNum, MonthNum, DayNum, HourNum ] = data_info( tiempo );
[wsdata, wstext,

```



```

wind_speed]=xlsread('./Input/Simulacion.xlsx','Wind_Speed');
[wsrow,wscolumn]=size(wind_speed);
%t=datevec(tiempo);
% month=t(:,2);
for c=1:wscolumn
if cell2mat(wind_speed(1,c))==MonthNum
for i=1:wscolumn
if strcmp(wind_speed(1,i),'time')
for k=2:wsrow-1
if
(HourNum>=(cell2mat(wind_speed(k,i))))&&(HourNum<cell2mat(wind_speed(k+1,
i)))
%la unitat de temps es prèn com a hora
%3 molins? 2 m de radi?
vProd(1,2)=0.5*1.226*pi*(cell2mat(wind_speed(k,c)))^3/3.6*3*2^2/1000;
quarter=floor(HourNum*4);
maxprod_data(1,2,quarter)=vProd(1,2);
break;
end
end
end
end
end
end
end
[srdata, srtext,
solar_radiation]=xlsread('./Input/Simulacion.xlsx','Solar_Radiation');
[srrow,srcolumn]=size(wind_speed);
%t=datevec(tiempo);
% month=t(:,2);
for c=1:srcolumn
if cell2mat(wind_speed(1,c))==MonthNum
for i=1:srcolumn
if strcmp(solar_radiation(1,i),'time')
for k=2:wsrow-1
if
(HourNum>=(cell2mat(solar_radiation(k,i))))&&(HourNum<cell2mat(solar_radi
ation(k+1,i)))
%la unitat de temps es prèn com a hora
%10 m2?
vProd(1,1)=(cell2mat(solar_radiation(k,c)))*10;
quarter=floor(HourNum*4);
maxprod_data(1,1,quarter)=vProd(1,1);
break;
end
end
end
end
end
end
end
%
% Identificar y Procesar eventuales nuevas órdenes
%
Filename='./input/PV_SimulationData.xml';
[WebValor,WebFecha,WebNombreTag]=LeeProcelWeb(Filename);
NCambios=size(WebNombreTag);
%Tenir en compte que si, després del balanç, els valors de les
%bateries (simulat i el de la optimització) no són iguals,
%és perquè l'energia generada no és igual i, per tant, cal que

```

```

%els valors reals siguin guardats a PV_SimulationData i
enviats
%a la ontologia per poder fer servir els valors a la
%optimització.
%
% Avanzar un instante de tiempo
%
tiempo=tiempo+t_paso;
vDistPlant= zeros(NProducts,NPlantas);
vDistMarkt= zeros(NProducts,NMarkets);
for Market=1:NMarkets
vDistPlant(:, :)= vDistPlant(:, :)+vDist(:, :,Market);
for Planta=1:NPlantas
vDistMarkt(:,Market)=
vDistMarkt(:,Market)+vDist(:,Planta,Market);
end
end
ProduccionAcumulada(:, :,Period)=ProduccionAcumulada(:, :,Period)+vProd(:, :
)*t_paso*24;
Storage_acumulada(:, :,Period)=Storage_acumulada(:, :,Period)+(vProd(:, :)+v
Ext(:, :)-vDistPlant(:, :))*t_paso*24;
DeliveryAcumulada(:, :,Period)=DeliveryAcumulada(:, :,Period)+vDistMarkt(:,
:)*t_paso*24;
vStrIn(:, :)=vProd(:, :)+vExt(:, :);
%
% Control automàtic: Ponerlo todo al máximo a menos que se hayan logrado
% los objetivos o se exceda la capacidad de almacenamiento
%
for Planta=1:NPlantas
for Product=1:NProducts
if Storage_acumulada(Product,Planta)<0
vDist(Product,Planta,:)=0;
else
vDist(Product,Planta,:)=DistCap(Product,Planta,:);
end
if
DeliveryAcumulada(Product,Market)>Demanda(Product,Market,Period)
vDist(Product,Planta,Market)=0;
end
end
end
end
%
%Función Objetivo
%
StorageCost= Storage_acumulada(1,1)*0.001;
ProductionCost=sum(sum(vProd(:, :)))*0.05*t_paso*24;
ConsumptionCost= sum(sum(vDistPlant(:, :)))*t_paso*24;
FOBJ=FOBJ+StorageCost+ProductionCost+ConsumptionCost;
FOBJ
Nivel_Bateria = Storage_acumulada(1,1)
% Otros càlculs para ProcelWeb
% Escritura de resutlados en XLM (primera version)
docNode=com.mathworks.xml.XMLUtils.createDocument('BXTHistoryLogType');
for P=1:NProducts
for Pl=1:NPlantas
for MaxProdPer=1:NMaxProdPeriod(2)
entry_node=docNode.createElement('RecordElement');

```

```

docNode.getDocumentElement.appendChild(entry_node);
%
ElementID_node=docNode.createElement('RecordElementID');
ElementID_text=docNode.createTextNode(nombre_maxprod_tag(P,Pl,MaxProdPer)
);
ElementID_node.appendChild(ElementID_text);
entry_node.appendChild(ElementID_node);
%
timestamp_node=docNode.createElement('LocalTime');
timestamp_text=docNode.createTextNode(datestr(tiempo,30));
timestamp_node.appendChild(timestamp_text);
entry_node.appendChild(timestamp_node);
%
storage_node=docNode.createElement('NewValue');
storage_text=docNode.createTextNode(num2str(maxprod_data(P,Pl,MaxProdPer)
));
storage_node.appendChild(storage_text);
entry_node.appendChild(storage_node);
end
end
end
for Product=1:NProducts
for Planta=1:NPlantas
for icase=1:4
entry_node=docNode.createElement('RecordElement');
docNode.getDocumentElement.appendChild(entry_node);
%
ElementID_node=docNode.createElement('RecordElementID');
switch icase
case 1
ElementID_text=docNode.createTextNode(Nombre_VelProd_Tag(Product,Planta))
;
case 2
ElementID_text=docNode.createTextNode(Nombre_VelStr_Tag(Product,Planta));
case 3
ElementID_text=docNode.createTextNode(Nombre_VelDistr_Tag(Product,Planta)
);
case 4
ElementID_text=docNode.createTextNode(Nombre_VelMkDist_Tag(Product,Planta)
));
otherwise
disp('desconocido')
end
ElementID_node.appendChild(ElementID_text);
entry_node.appendChild(ElementID_node);
%
timestamp_node=docNode.createElement('LocalTime');
timestamp_text=docNode.createTextNode(datestr(tiempo,30));
timestamp_node.appendChild(timestamp_text);
entry_node.appendChild(timestamp_node);
%
storage_node=docNode.createElement('NewValue');
switch icase
case 1
%
storage_text=docNode.createTextNode(num2str(Pro
% duccionAcumulada(Product,Planta)));
storage_text=docNode.createTextNode(num2str(vProd(Product,Planta)));

```

```

case 2
storage_text=docNode.createTextNode(num2str(vStrIn(Product,Planta)));
case 3
storage_text=docNode.createTextNode(num2str(vDist(Product,Planta)));
case 4
%
storage_text=docNode.createTextNode(num2str(vSt
% rIn(Product,Planta)));
storage_text=docNode.createTextNode(num2str(vDistMarkt(Product,Planta)));
case 5
for MaxProdPeriod=1:NMaxProdPeriod
ElementID_text=docNode.createTextNode(nombre_maxprod_tag(Product,Planta,M
axProdPeriod));
end
otherwise
disp('desconocido')
end
storage_node.appendChild(storage_text);
entry_node.appendChild(storage_node);
%-----
end
end
for Market=1:NMarkets
for icafe=1:4
entry_node=docNode.createElement('RecordElement');
docNode.getDocumentElement.appendChild(entry_node);
%
ElementID_node=docNode.createElement('RecordElementID');
switch icafe
case 1
ElementID_text=docNode.createTextNode(Nombre_VelMkDist_Tag(Product,Market
));
case 2
ElementID_text=docNode.createTextNode(Nombre_TotDist_Tag(Product,Market)
);
case 3
ElementID_text=docNode.createTextNode(nombre_ini_tag(Product,Market));
case 4
ElementID_text=docNode.createTextNode(nombre_dur_tag(Product,Market));
end
ElementID_node.appendChild(ElementID_text);
entry_node.appendChild(ElementID_node);
%
timestamp_node=docNode.createElement('LocalTime');
timestamp_text=docNode.createTextNode(datestr(tiempo,30));
timestamp_node.appendChild(timestamp_text);
entry_node.appendChild(timestamp_node);
%
storage_node=docNode.createElement('NewValue');
if icafe==1
storage_text=docNode.createTextNode(num2str(vDistMarkt(Product,Market)));
else
storage_text=docNode.createTextNode(num2str(DeliveryAcumulada(Product,Mar
ket)));
end
storage_node.appendChild(storage_text);
entry_node.appendChild(storage_node);
%-----

```

```
end
end
end
entry_node=docNode.createElement('RecordElement');
docNode.getDocumentElement.appendChild(entry_node);
%
ElementID_node=docNode.createElement('RecordElementID');
ElementID_text=docNode.createTextNode('Real_TotalCost');
ElementID_node.appendChild(ElementID_text);
entry_node.appendChild(ElementID_node);
%
timestamp_node=docNode.createElement('LocalTime');
timestamp_text=docNode.createTextNode(datestr(tiempo,30));
timestamp_node.appendChild(timestamp_text);
entry_node.appendChild(timestamp_node);
%
fobj_node=docNode.createElement('NewValue');
fobj_text=docNode.createTextNode(num2str(FOBJ));
fobj_node.appendChild(fobj_text);
entry_node.appendChild(fobj_node);
% string01=datestr(tiempo,13);
% string01(3)='_';
% string01(6)='_';
%
filename=strcat('./Output/PV_DynamicData',string01,'.xml');
filename=strcat('./Output/PV_DynamicData.xml');
xmlwrite(filename,docNode)
%
% Escritura OPC
%
% info = opcserverinfo('localhost')
% da = opcda('localhost', 'GrayBox.OPC.Competitive');
% connect(da);
% allIDs = getIDList(da.ItemAttributes)
% write(GObj,FOBJ);
pause(t_wait);
end
% El storage se queda igual al cambiar de periodo; La producción
acumulada y la delivery acumulada se reinician
ProduccionAcumulada=zeros(NPlantas,NProducts);
DeliveryAcumulada=zeros(NProducts,NMarkets);
end
```